

Proyecto fin de carrera

Adaptación e implantación de un sistema ERP Libre en una ONG

versión v.1.2

07 de Junio 2009



Autor:	Gómez Díaz, Diego

Tutora:	Fraga Vázquez, Anabel

Índice

ÍNDICE	I
ILUSTRACIONES	III
REQUISITOS DE USUARIO	IV
REQUISITOS NO FUNCIONALES.....	V
REQUISITOS DE SOFTWARE	V
INTRODUCCIÓN	1
1 RESUMEN	1
2 OBJETIVOS.....	2
3 ENTIDAD DESTINATARIA.....	2
3.1 <i>Relación del autor con la entidad</i>	4
3.2 <i>Motivación</i>	5
3.2.1 <i>Problemática</i>	5
ESTADO DEL ARTE	7
1 ENTERPRISE RESOURCE PLANNING (ERP)	7
2 CUSTOMER RELATIONSHIP MANAGEMENT (CRM).....	8
3 TECNOLOGÍAS ESTUDIADAS	9
3.1 <i>Alternativas</i>	9
3.2 <i>Solución a Aplicar</i>	10
4 LA HERRAMIENTA	12
4.1 <i>Características</i>	12
4.2 <i>Arquitectura</i>	13
4.3 <i>Soporte</i>	14
4.4 <i>Documentación</i>	15
ANÁLISIS (DISEÑO ARQUITECTÓNICO)	16
1 REQUISITOS DE USUARIO	16
1.1 <i>Requisitos Funcionales</i>	16
1.2 <i>Requisitos No Funcionales</i>	23
2 MODELO CONCEPTUAL.....	26
3 ARQUITECTURA DEL SISTEMA	27
4 REQUISITOS DE SOFTWARE.....	29
DISEÑO DETALLADO.....	43
1 SISTEMA.....	43
1.1 <i>Convenio de diseño</i>	43
1.2 <i>Módulos nuevos</i>	44
1.2.1 <i>Gestión de Áreas</i>	44
1.2.2 <i>Gestión de Proyectos</i>	45
1.2.3 <i>Gestión de Programas</i>	46
1.2.4 <i>Gestión de Actividades</i>	47
1.2.5 <i>Gestión de Inscripciones en Programas y Actividades</i>	48
1.2.6 <i>Generación de Recibos de Cobro</i>	53
1.2.7 <i>Notificación por SMS</i>	55
1.2.8 <i>Informes de Mailing</i>	58
1.3 <i>Funcionalidad cambiada o expandida del ERP</i>	59
1.3.1 <i>Compatibilidad con sistema Openbravo ERP</i>	59
1.3.2 <i>Gestión de Participantes</i>	59
1.3.3 <i>Nueva vista de Fotografías de Usuarios</i>	61
1.3.4 <i>Atributos Personalizados</i>	63
1.4 <i>Importación de los datos existentes</i>	65

1.4.1	Análisis de los sistemas heterogéneos.....	65
1.4.2	Definición de la correlación de datos.....	67
1.4.3	Unificación de fuentes de datos.....	67
1.4.4	Creación del proceso de Migración	68
2	DESARROLLO DEL APLICATIVO	69
2.1	<i>Desarrollo en entorno de desarrollo.....</i>	69
2.2	<i>Pruebas en entorno de test.....</i>	69
2.3	<i>Pruebas en entorno productivo.....</i>	70
2.4	<i>Corrección de errores.....</i>	70
3	IMPLANTACIÓN DEL SISTEMA	71
3.1	<i>Fases de implantación</i>	71
3.2	<i>Procedimiento.....</i>	72
3.2.1	Creación de repositorio de control de versiones.....	72
3.2.2	Instalación del sistema en entorno productivo.....	72
3.2.3	Configuración del sistema de backups.....	75
3.2.4	Instalación del sistema de gestión de incidencias	76
3.2.5	Instalación del entorno de test	77
3.2.6	Importación de datos	77
4	PROGRAMA FORMATIVO.....	80
4.1	<i>Antecedentes</i>	80
4.2	<i>Diseño de curso SCORM.....</i>	80
4.3	<i>Curso formativo presencial</i>	81
PLANIFICACIÓN		83
1	PLANIFICACIÓN INICIAL	83
2	PLANIFICACIÓN FINAL	83
3	PRESUPUESTO.....	86
3.1	<i>Licencias de Software</i>	86
3.2	<i>Recursos humanos.....</i>	86
3.3	<i>Coste de hardware</i>	87
3.4	<i>Balance final del presupuesto</i>	87
RESULTADOS.....		88
1	RESULTADOS ALCANZADOS.....	88
2	PROYECTOS SIMILARES	89
3	LÍNEAS FUTURAS DE TRABAJO.....	90
4	CONCLUSIONES	91
REFERENCIAS		93
ANEXO I – MANUAL DE USUARIO.....		95
1	ORGANIZACIÓN DEL CURSO	95
1.1	<i>Objetivos.....</i>	95
1.2	<i>Ejercicios y material del curso.....</i>	95
2	¿POR QUÉ YMCA OPENBRAVO?.....	96
2.1	<i>Necesidades y Soluciones.....</i>	96
2.2	<i>Descripción del proyecto.....</i>	96
2.3	<i>Autor.....</i>	96
3	INTRODUCCIÓN A YMCA OPENBRAVO.....	97
3.1	<i>Prerrequisitos.....</i>	97
3.2	<i>Acceso a la aplicación y notas sobre seguridad.....</i>	98
3.3	<i>Login/Logout.....</i>	99
3.4	<i>Cambio de Contraseña.....</i>	100
3.5	<i>Cambio de Rol</i>	101
3.6	<i>Interfaz</i>	102
3.6.1	Menú de herramientas	102
3.6.2	Controles de navegación.....	103
3.6.3	Referencias entre registros	105
3.7	<i>Estructura y nomenclatura de la aplicación.....</i>	106

3.8	Herramientas de gestión.....	106
3.9	Herramienta de búsqueda	107
3.10	Informes y listados.....	107
4	PROCESOS HABITUALES	108
4.1	Alta de participante.....	108
4.2	Alta de proyectos, programa y actividades.....	113
4.3	Inscripción de participante.....	115
4.4	Generación de recibos	116
4.5	Envío de SMS a participantes.....	118
5	HERRAMIENTAS ADMINISTRATIVAS.....	119
5.1	Alta de usuario.....	119
5.2	Baja de usuario.....	123
5.3	Gestión de roles.....	123
5.4	Configuración del servicio de SMS.....	124
6	REPORTE DE ERRORES.....	125
6.1	Necesidad	125
6.2	Procedimiento.....	125
7	AMPLIANDO EL CONOCIMIENTO	127
7.1	Ayuda de la aplicación.....	127
7.2	Curso auto-formativo	127
ANEXO II – PROCESO DETALLADO DE MIGRACIÓN.....		128
1	MIGRACIÓN AL SISTEMA PRODUCTIVO.....	128
2	MIGRACIÓN AL SISTEMA DEMOSTRATIVO	141
ANEXO III - CONTENIDO DEL DVD.....		143
1	CONTENIDO DEL DVD	143
2	EJECUCIÓN DEL ENTORNO DEMOSTRATIVO	144
3	CURSO ONLINE.....	148

Ilustraciones

ILUSTRACIÓN 1:	DEPARTAMENTOS DE YMCA ESPAÑA.....	3
ILUSTRACIÓN 2:	ORGANIGRAMA ASOCIACIÓN YMCA	4
ILUSTRACIÓN 3:	DESPLIEGUE OPENBRAVO	12
ILUSTRACIÓN 4:	ARQUITECTURA OPENBRAVO	13
ILUSTRACIÓN 5:	MODELO CONCEPTUAL DE LOS ELEMENTOS CLAVES DEL SISTEMA	26
ILUSTRACIÓN 6:	MODELO CONCEPTUAL TIPOS DE PERSONA.....	26
ILUSTRACIÓN 7:	ARQUITECTURA YMCA OPENBRAVO	27
ILUSTRACIÓN 8:	MODELO DE DATOS DEL ÁREA	44
ILUSTRACIÓN 9:	INTERFAZ DE INSERCIÓN DE ÁREAS	44
ILUSTRACIÓN 10:	INTERFAZ DE INSERCIÓN DE PROYECTOS	45
ILUSTRACIÓN 11:	MODELO DE DATOS DE PROYECTO	45
ILUSTRACIÓN 12:	INTERFAZ DE INSERCIÓN DE PROGRAMAS.....	46
ILUSTRACIÓN 13:	MODELO DE DATOS DE PROGRAMAS Y ACTIVIDADES	47
ILUSTRACIÓN 14:	INTERFAZ DE INSERCIÓN DE ACTIVIDADES	48
ILUSTRACIÓN 15:	INTERFAZ DE INSCRIPCIÓN DE PARTICIPANTES	49
ILUSTRACIÓN 16:	LISTADO DE INSCRIPCIONES	49
ILUSTRACIÓN 17:	CLASES QUE IMPLEMENTA EL PROCESO DE BACKGROUND	50
ILUSTRACIÓN 18:	CONTROL DEL PROCESO EN BACKGROUND.....	50
ILUSTRACIÓN 19:	MODELO DE DATOS DE LAS INSCRIPCIONES.....	51
ILUSTRACIÓN 20:	CALLOUT PARA LA ACTUALIZACIÓN DE PRECIOS.....	52
ILUSTRACIÓN 21:	INTERFAZ DE EDICIÓN DE RECIBOS GENERADOS	53
ILUSTRACIÓN 22:	FORMATO DEL RECIBO GENERADO	54

ILUSTRACIÓN 23: INTERFAZ DE ENVÍO DE MENSAJES DE MÓVILES	55
ILUSTRACIÓN 24: MODELO MÓDULO DE ENVÍO DE SMS	56
ILUSTRACIÓN 25: CLASE DE ENVÍO DE SMS	56
ILUSTRACIÓN 26: CLASES GATEWAY DE ENVÍO DE SMS	57
ILUSTRACIÓN 27: INTERFAZ DE FILTRO DE CREACIÓN DE ETIQUETAS DE MAILING	58
ILUSTRACIÓN 28: INTERFAZ DE EDICIÓN DE PARTICIPANTES	60
ILUSTRACIÓN 29: INCLUSIÓN DE FOTOGRAFÍA	61
ILUSTRACIÓN 30: DIAGRAMA DE CLASES DE LOS CONTROLADORES WAD AÑADIDOS	62
ILUSTRACIÓN 31: CONTROLADOR DE CARGA DE FOTOGRAFÍAS	62
ILUSTRACIÓN 32: CREACIÓN DE ATRIBUTO	63
ILUSTRACIÓN 33: CREACIÓN DE CONJUNTO DE ATRIBUTOS	63
ILUSTRACIÓN 34: EDICIÓN DE ATRIBUTO EN INSCRIPCIÓN DE PARTICIPANTE	64
ILUSTRACIÓN 35: SISTEMA EXISTENTE DESDE EL QUE SE QUIERE MIGRAR	65
ILUSTRACIÓN 36: MODELO DE DATOS BASE DATOS ACCESS	66
ILUSTRACIÓN 37: METODOLOGÍA DE DESARROLLO	69
ILUSTRACIÓN 38: ROLES DE USUARIO	73
ILUSTRACIÓN 39: BACKUP DEL SISTEMA	75
ILUSTRACIÓN 40: IMPORTAR PARTICIPANTES DESDE LAS TRES FUENTES DE DATOS EXISTENTES	78
ILUSTRACIÓN 41: IMPORTACIÓN DESDE ARCHIVO EXCEL	78
ILUSTRACIÓN 42: MAPEO DE DATOS EN TALEND STUDIO	79
ILUSTRACIÓN 43: CURSO SCORM DESPLEGADO EN LA PLATAFORMA MOODLE	81
ILUSTRACIÓN 44: CRONOGRAMA DEL PROYECTO	84
ILUSTRACIÓN 45: DEPENDENCIA ENTRE ELEMENTOS DEL SISTEMA	106
ILUSTRACIÓN 46: CURSO ONLINE SIN ENTORNO DE APRENDIZAJE	148

Requisitos de usuario

RF.1 - GESTIÓN DE PARTICIPANTES, VOLUNTARIOS, TRABAJADORES	16
RF.1.1 - INCLUIR FOTOGRAFÍA DE PARTICIPANTE	16
RF.1.2 - INFORMACIÓN ACADÉMICA	17
RF.1.3 - COMUNICACIONES A PETICIÓN	17
RF.1.4 - ADJUNTAR ARCHIVOS	17
RF.2 - GESTIÓN DE ÁREAS	17
RF.3 - GESTIÓN DE PROYECTOS	17
RF.3.1 - DOCUMENTACIÓN ADJUNTA AL PROYECTO	17
RF.4 - GESTIÓN DE PROGRAMAS	18
RF.4.1 - DEFINICIÓN DE NUEVOS PARÁMETROS	18
RF.5 - GESTIÓN DE ACTIVIDADES	18
RF.6 - PRECIO DE SOCIO	18
RF.7 - INSCRIPCIÓN DE PARTICIPANTES	18
RF.8 - IMPUTACIÓN DE GASTOS	19
RF.9 - ELABORACIÓN DE RECIBOS	19
RF.9.1 - NUMERACIÓN ÚNICA DE RECIBOS	19
RF.9.2 - RECIBOS IMPAGADOS	19
RF.9.3 - GENERACIÓN AUTOMÁTICA DE RECIBOS	19
RF.9.4 - FACTURA DE SERVICIO	19
RF.10 - INTRODUCCIÓN DE FACTURAS DE COMPRA	20
RF.11 - GESTIÓN DE SEDES	20
RF.12 - GESTIÓN DE USUARIOS	20
RF.13 - GESTIÓN DE ROLES	20
RF.14 - CAPACIDAD DE COMUNICACIÓN POR SMS	20
RF.L.1 - LISTADO DE PARTICIPANTES DE PROGRAMA	21



RF.L.2 - FICHA RESUMEN DE PROGRAMA	21
RF.L.3 - LISTADO DE PARTICIPANTES DE ACTIVIDAD	21
RF.L.4 - FICHA RESUMEN DE ACTIVIDAD	21
RF.L.5 - LISTADOS DE PROYECTOS POR ÁREA	21
RF.L.6 - LISTADOS DE PROGRAMAS POR PROYECTO	21
RF.L.7 - INFORME DE PARTICIPACIÓN	22
RF.L.8 - LISTADOS DE PARTICIPANTES POR ÁREAS DE INTERÉS	22
RF.L.9 - LISTADO DE VOLUNTARIOS	22
RF.L.10 - LISTADOS EN FORMATOS ESPECIALES, MAILING	22
RF.L.11 - FICHA RESUMEN DE PARTICIPANTE	22
RF.L.12 - LISTADO DE MAILING	22

Requisitos no funcionales

RNF.1 - ACCESO SIMULTANEO	23
RNF.2 - ACCESO DISTRIBUÍDO	23
RNF.3 - MULTISISTEMA	23
RNF.4 - DISPONIBILIDAD DURANTE JORNADA LABORABLE	23
RNF.5 - SISTEMA INTUITIVO Y AUTOFORMATIVO	23
RNF.6 - AYUDA DISPONIBLE	24
RNF.7 - GARANTIZAR AUTENTICACIÓN	24
RNF.8 - GARANTIZAR LA MANTENIBILIDAD DEL SOFTWARE	24
RNF.9 - SISTEMA MULTIIDIOMA	24
RNF.10 - BACKUP AUTOMATIZADO	24
RNF.11 - CAPAZ DE FUNCIONAR CON LOS RECURSOS DISPONIBLES	24
RNF.12 - MIGRACIÓN DE DATOS	25
RNF.13 - INDEPENDENCIA DEL SERVIDOR DE SMS	25
RNF.14 - MINIMIZAR COSTES DE COMUNICACIÓN	25

Requisitos de Software

SR.1 - ALTA DE PARTICIPANTES, VOLUNTARIOS, TRABAJADORES	29
SR.2 - INCLUIR FOTOGRAFÍA DE PARTICIPANTE	29
SR.3 - INFORMACIÓN ACADÉMICA	30
SR.4 - COMUNICACIONES A PETICIÓN	30
SR.5 - ADJUNTAR ARCHIVOS	30
SR.6 - BAJA DE PARTICIPANTES	30
SR.7 - GESTIÓN DE ÁREAS	30
SR.8 - ALTA DE PROYECTOS	31
SR.9 - DOCUMENTACIÓN ADJUNTA AL PROYECTO	31
SR.10 - BAJA DE PROYECTOS	31
SR.11 - ALTA DE PROGRAMAS	32
SR.12 - BAJA DE PROGRAMA	32
SR.13 - DEFINICIÓN DE ATRIBUTOS	32
SR.14 - DEFINICIÓN DE CONJUNTOS DE ATRIBUTOS	33
SR.15 - ALTA DE ACTIVIDADES	33

SR.16 - BAJA DE ACTIVIDADES.....	33
SR.17 - PRECIO DE SOCIO	34
SR.18 - INSCRIPCIÓN DE PARTICIPANTES	34
SR.19 - CANCELACIÓN DE INSCRIPCIÓN	34
SR.20 - ELABORACIÓN DE RECIBOS.....	34
SR.21 - NUMERACIÓN ÚNICA DE RECIBOS	34
SR.22 - RECIBOS IMPAGADOS	35
SR.23 - GENERACIÓN AUTOMÁTICA DE RECIBOS	35
SR.24 - GESTIÓN DE SEDES.....	35
SR.25 - ALTA DE USUARIOS	35
SR.26 - BAJA DE USUARIOS	36
SR.27 - ALTA DE ROLES	36
SR.28 - BAJA DE ROL	36
SR.29 - LOGIN EN EL SISTEMA	36
SR.30 - CAMBIO DE ROL	36
SR.31 - ALTA DE ENVÍO DE SMS.....	37
SR.32 - ALTA MANUAL DE DESTINATARIOS SMS.....	37
SR.33 - BAJA DE DESTINATARIOS SMS	37
SR.34 - ALTA DE CUENTAS DE ENVÍO DE SMS	37
SR.35 - ALTA DE NUEVOS SERVICIOS DE ENVÍO DE SMS.....	38
SR.36 - UNIFICACIÓN DE DESTINATARIOS	38
SR.37 - HISTORIAL DE ENVÍO DE SMS.....	38
SR.38 - LISTADO DE PARTICIPANTES DE PROGRAMA	38
SR.39 - FICHA RESUMEN DE PROGRAMA	38
SR.40 - LISTADO DE PARTICIPANTES DE ACTIVIDAD	39
SR.41 - FICHA RESUMEN DE ACTIVIDAD	39
SR.42 - LISTADOS DE PROYECTOS POR ÁREA.....	39
SR.43 - LISTADOS DE PROGRAMAS POR PROYECTO	39
SR.44 - INFORME DE PARTICIPACIÓN	39
SR.45 - LISTADOS DE PARTICIPANTES POR ÁREAS DE INTERÉS.....	40
SR.46 - LISTADO DE VOLUNTARIOS	40
SR.47 - LISTADOS EN FORMATOS ESPECIALES, MAILING.....	40
SR.48 - MAILING FILTRADO	40
SR.49 - FICHA RESUMEN DE PARTICIPANTE	40
SR.50 - LISTADO DE MAILING	40
SR.51 - SCRIPTS DE MIGRACIÓN	41
SR.52 - MULTIIDIOMA	41
SR.53 - COMPATIBILIDAD HACIA ATRÁS	41
SR.54 - ARQUITECTURA WEB	41
SR.55 - SISTEMA MULTISESIÓN	41
SR.56 - SISTEMA DE AYUDA.....	42
SR.57 - CURSO ONLINE	42
SR.58 - INTERFAZ AMIGABLE	42
SR.59 - BACKUP AUTOMATIZADO	42

Introducción

1 Resumen

El proyecto que se describe en los próximos capítulos fue realizado como proyecto de fin de carrera para la licenciatura de Ingeniería superior de Informática.

El proyecto se centra en ofrecer una solución a la problemática que existía en una organización sin ánimo de lucro, que tenía problemas en su coordinación y gestión a nivel nacional por su falta de informatización y unicidad en los procedimientos.

Con este pretexto se desarrolla el proyecto, que a la vez, pretende cubrir las etapas principales de la vida de un producto de Software.

Así el proyecto cubre las siguientes fases:

- Estudio de la problemática: en la que se describe y analiza la situación de la organización que insta a la realización de una solución de estas características.
- Estudio de alternativas de mercado: de aplicaciones de gestión que ofrezcan una solución eficaz, sencilla y mantenible tras la finalización del proyecto y el periodo de mantenimiento.
- Análisis en profundidad de la solución esperada por la organización.
- Diseño de la plataforma
- Implementación del sistema: adaptación del sistema existente e implementación de nuevos módulos que no cubra la solución.
- Creación de entorno de pruebas e integración
- Instalación y mantenimiento del sistema productivo
- Seguimiento y resolución de incidencias
- Documentación y elaboración de cursos formativos del sistema.

Todas estas fases son cubiertas gracias a la experiencia desarrollada en algunas de las asignaturas de la titulación. Desde las que cubren los procesos de ingeniería de Software, a las más técnicas, a lo largo de este proceso se desarrollan muchas habilidades multidisciplinares que pretenden lograr un resultado lo más completo y profesional posible.

En el siguiente apartado describimos las características más relevantes de la herramienta que hemos desarrollado para la práctica de detección del plagio.

2 Objetivos

Ofrecer solución a la problemática de gestión de la organización, garantizando la independencia del desarrollo y su continuidad futura tras la finalización del proyecto.

Elaborar un proyecto real que cubra las fases principales de la vida de un software y donde se desarrollen las habilidades adquiridas durante algunos cursos de la carrera de Ingeniería Informática.

3 Entidad destinataria

YMCA (Young Men's Christian Association) es una de las organizaciones internacionales de carácter voluntario y no lucrativo más extendidas del mundo. Desde que fuera fundada en 1844 en Londres por un grupo de jóvenes encabezados por George Williams, YMCA ha trabajado por la unidad, con dos propósitos principales: el entendimiento entre los pueblos y su contribución a una sociedad basada en la justicia y la paz así como en el desarrollo integral y armónico del ser humano.

A lo largo de los años se ha reconocido la valía de la labor de la asociación. En 1946 el presidente de la Alianza Mundial de YMCAs, el Dr. John R. Mott, recogió el Premio Nobel de la Paz que se otorgaba a la YMCA por las labores humanitarias realizadas en ambos lados del frente durante las Guerras Mundiales.

Deportes como el baloncesto, el fútbol sala y el volley ball fueron creados en centros de YMCA por parte de educadores y profesores de educación física de la Asociación.

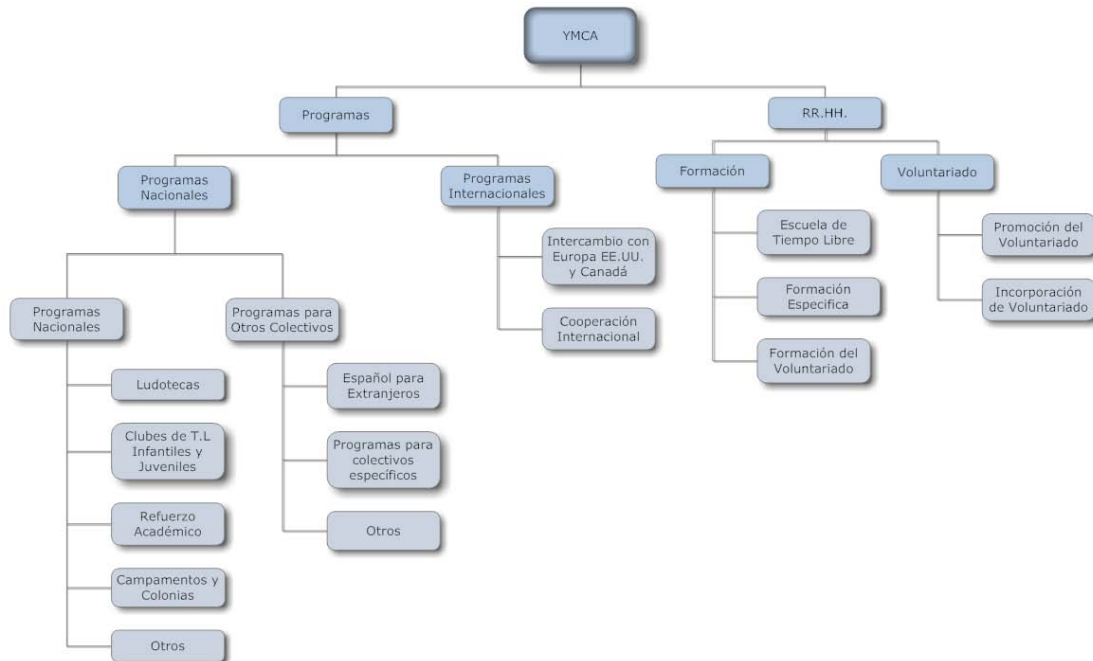


Ilustración 1: Departamentos de YMCA España

En España, YMCA(1) comenzó su andadura en 1980 en barrios periféricos de Madrid, como respuesta a las necesidades de los y las jóvenes y con el objetivo básico del desarrollo de la infancia y la juventud más necesitada. En 1980 se produjo el reconocimiento internacional de YMCA España, así como su declaración de Asociación de Utilidad Pública. En la actualidad YMCA cuenta con diecisiete centros de programas en Madrid, Getafe, Leganés, Barcelona, Valencia, Zaragoza, Huesca, Toledo, Salamanca, Valladolid y La Rioja, donde numerosos grupos de niños, niñas y jóvenes se reúnen para realizar todo tipo de actividades deportivas, culturales, ludotecas, recreativas, formativas, de tiempo libre y de cooperación internacional.

YMCA está presente en más de 125 países y cuenta con más de 40 millones de miembros, a los que ofrece un gran número de servicios, así como albergues, campamentos e instalaciones deportivas y recreativas. YMCA es órgano consultivo ante el Consejo Económico y Social de la ONU.

3.1 Relación del autor con la entidad

La elaboración del proyecto para la entidad YMCA España no ha sido casual, sino una oportunidad que el autor de este proyecto quería aprovechar volcando sus conocimientos técnicos en un proyecto que beneficiase a la asociación con la que desde el año 2004 ha estado trabajando como voluntario en diferentes áreas.

El comienzo de la trayectoria en la asociación comenzó en 2004 participando en programas de apoyo escolar y ocio dirigidos a la infancia. Posteriormente ha participado en otros programas nacionales e internacionales, como encuentros o campamentos.

Desde hace dos años forma parte de uno de los consejos locales que pretenden aportar su experiencia a asesorar a los responsables de cada región.

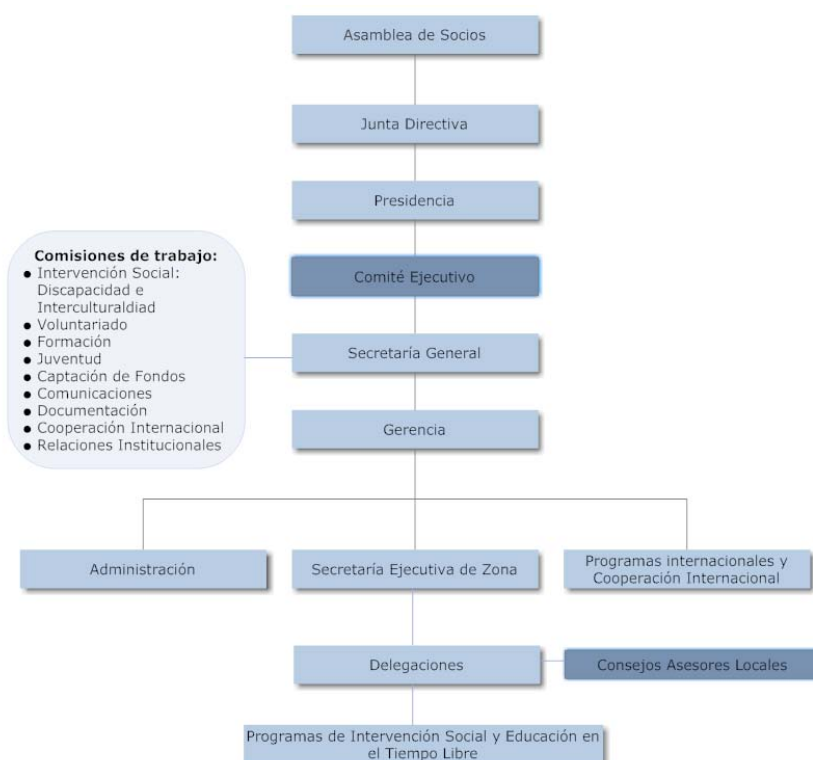


Ilustración 2: Organigrama asociación YMCA

Desde finales del año 2007 fue invitado a formar parte del comité ejecutivo de la asociación, que bajo un puesto de voluntario, ha podido ayudar a tomar decisiones más trascendentes dentro de la asociación.

Esta presencia en diferentes áreas de la asociación le ha permitido obtener una visión más amplia de la problemática interna y ofrecer una solución que contribuya a su desarrollo futuro.

3.2 Motivación

Durante la última semana del mes de Mayo del 2008 se mantuvo una reunión informal con el gerente de la asociación para tratar de determinar cuáles eran las problemáticas más relevantes producidas por la falta de un sistema de información adecuado. Con el objetivo de buscar una solución para construir un sistema base que pudiera evolucionar una vez concluido este proyecto, y que en primera instancia pudiera dar solución a los elementos que estaban dificultando el crecimiento de la asociación.

3.2.1 Problemática

En los 28 años que tiene de antigüedad la asociación, ha sufrido altibajos que si bien le permitió mantener su posición en diferentes sectores, y estar presente en diferentes provincias Españolas, no ha sido hasta los últimos 5 años cuando ha comenzado a experimentar un crecimiento significativo.

Este crecimiento se produjo tras una reestructuración interna que cambió muchos de los procesos internos de la asociación, a nivel administrativo y jerárquico, para tener un control centralizado de todas las sedes repartidas por el territorio nacional.

Sin embargo los sistemas informáticos utilizados hasta el momento se mantuvieron o adaptaron a las nuevas necesidades, y aunque en un principio esto no supuso un gran inconveniente hoy en día lo es.

En cuestión de 5 años se han cuadruplicado el presupuesto que se maneja a nivel nacional y se ha duplicado la plantilla. La toma de decisiones no se puede apoyar en los balances económicos ya que estos, por culpa de los procesos actuales, se conocen con un retardo de varios meses.

Por otro lado los procesos actuales no permiten hacer una contabilidad analítica ni permite asignar recursos a los procesos para llevar un control adecuado. Esto es de especial importancia en una asociación como esta que trabaja con multitud de subvenciones que, aparte de estar bien justificadas, deben de aplicarse adecuadamente a los proyectos, manteniendo la compatibilidad entre diferentes subvenciones y evitando que dos



subvenciones diferentes se apliquen a un mismo proyecto por duplicado. Situaciones que podrían forzar la devolución de una subvención y hacer peligrar la contabilidad general de la asociación.

A parte es necesario mejorar otros procesos de trabajo en red, para que toda la asociación pueda trabajar sobre criterios unificados, tengan un registro unificado de los participantes, voluntarios y entidades colaboradoras, y que de alguna forma puedan también compartir los recursos de que disponen (proyectos, recursos humanos, cursos formativos...).

Estado del arte

En este apartado repasamos algunos fundamentos sobre los que se basa este proyecto, para establecer las bases que permitan una lectura adecuada de esta memoria por alguien ajeno a estas tecnologías.

1 Enterprise Resource Planning (ERP)

Toda empresa u organización que quiera hoy en día competir y sobrevivir en el mercado debe conocer lo que está haciendo en tiempo real para tratar de solventar los problemas que surjan dentro de la entidad, y buscar nuevas formas de trabajo ante la realidad cambiante.

Las empresas al comenzar su andadura centran su atención en su idea de negocio que es la que le da forma. A menudo por falta de experiencia, formación o personal suficiente, la gestión queda relegada a segundo plano y esta se vuelve bastante pobre en muchas pequeñas y medianas empresas. A lo largo de la última década han sido cada vez más las empresas que han visto en la informática y los paquetes de gestión una herramienta útil que les permitía tener una visión más actualizada del estado de su negocio.

Sin embargo la mayoría de estos paquetes resultan interesantes cuando la empresa o entidad es de un tamaño pequeño y su funcionamiento es muy homogéneo. Pero es to raramente ocu rre, y finalmente las em presas continúan creciendo y su operativa queda distribuida en diferentes departamentos cuyas necesidades informáticas son diferentes, como consecuencia comienzan a surgir muchos sistemas heterogéneos y la organización pasa a contar con mucha información pero mal distribuido, de la que es difícil obtener conclusiones generales.

Para solucionar esta problemática, las grandes corporaciones han estado usando sistemas ERP como SAP o Microsoft Navision que son sistemas adaptados capaces de aunar todas estas aplicaciones dirigidas a cada actividad empresarial, en una sola aplicación, que mantiene controlados y relacionados toda la información. De esta forma los ge stores son capaces de estar bien comunicados con el resto de departamentos y compartir la información útil y necesaria, a la vez que es posible obtener una fotografía en tiempo real de lo que ocurre en la organización gracias a ciertas herramientas (Business Intelligence).

En el momento en que crece ya es necesario recurrir a sistemas más complejos y caros que muchas empresas pequeñas no pueden permitirse.

Las aplicaciones Enterprise Resource Planning (ERP en adelante), permite dar solución a los diferentes departamentos de una empresa de una forma adaptada a sus necesidades y mantener todos los datos generados para poder extraer información útil que permita mantener el volumen de crecimiento.

Así los ERP hay que verlos también como una herramienta imprescindible que permite que las empresas puedan crecer y superar lo que se le escapa del control de una persona con su capacidad y visión sin ayuda de medios telemáticos.

2 Customer Relationship Management (CRM)

Otro enfoque que eligen otras organizaciones, y que a menudo complementa al ERP es el uso de Customer Relationship Management Software (ó CRM). Este tipo de sistemas permite organizar los procesos empresariales en torno a los clientes, almacenando todo el historial del cliente con la empresa, para permitir ofrecer un servicio más personalizado y obtener ventajas estratégicas en base a la información que estos sistemas pueden sintetizar a partir de todo el bulto de datos almacenado.

Puesto que una de las claves del proyecto era la gestión de los participantes, este podría haber sido un buen punto de partida para la creación de un sistema más enfocado a los clientes, y menos a la gestión financiera.

En una u otra dirección estos sistemas suelen comunicarse con herramientas ERP, para complementar e intercambiar información útil para cada uno de los sistemas.

A nivel empresarial algunos de los productos propietarios más fuertes son Siebel de Oracle, o SAP CRM. Sin embargo también existen alternativas opensource muy potentes como SugarCRM(2).

3 Tecnologías estudiadas

Para encontrar la mejor solución para afrontar el problema al que se enfrentaba la organización se estudiaron varias posibilidades que se exponen a continuación.

3.1 Alternativas

La asociación YMCA España, ha experimentado un crecimiento en volumen de usuarios, empleados y programas, y hace tiempo ya se planteó la necesidad de instaurar un sistema de gestión integral para toda la asociación. Así se comenzó a estudiar algunas posibilidades.

Inicialmente se evaluaron las experiencias de otras entidades similares, como Cáritas, Cruz Roja o ASDE. Estas entidades han dado el paso de implantar sistemas informáticos en diferentes medidas y sin embargo ninguno ha obtenido una experiencia lo suficientemente positiva para seguir sus pasos.

El mayor problema radica en la complejidad de este tipo de infraestructuras y la falta de personal cualificado que se encargue de su mantenimiento. Dentro de las acciones sociales de algunas empresas consultoras está la de destinar horas de consultoría a organizaciones sin ánimo de lucro, y por lo tanto algunas han recibido el asesoramiento y soporte de grandes empresas. Pero es tan importante la implantación y adaptación con su mantenimiento posterior.

La organización al no tener muy claro cuál podría ser su mejor alternativa consultó a varias empresas que le ofrecieron soluciones diversas pero muy caras y complejas. Como Microsoft Navision, que aparte de ser un sistema muy caro, no se adaptaba bien a las necesidades, por ser demasiado grande y requerir de una adaptación aún más costosa.

Tras la reunión mantenida en Mayo con el gerente de la asociación, se plantearon varias alternativas para dar solución a las necesidades básicas.

Una primera solución sería el crear un sistema desde cero que contemplase los requisitos más relevantes y tuviese en cuenta el tiempo disponible para hacer el proyecto y la dificultad asociada. Esta solución implicaba sin embargo muchos inconvenientes: para empezar el número de requisitos no podría ser muy elevado, puesto que al tratar con procesos contables, cada funcionalidad debería estar bien diseñada, desarrollada y probada. En segundo lugar, un sistema personalizado como este no permitiría en un futuro adaptar el sistema a los cambios con facilidad.

La segunda solución, consistiría en partir de un sistema existente, que se adaptase a las necesidades según los requisitos y se desarrollasen los módulos necesarios. Esto reduciría

el tiempo de desarrollo y permitiría abarcar mayor número de requisitos. Además quedaría garantizada la vida del software si otras empresas pudieran mantener el sistema, que al tratarse de un sistema comercial estaría soportado por especialistas capaces de retomar un proyecto ya instaurado.

Finalmente se optaría por esta segunda opción puesto que garantizar la continuidad del sistema una vez el periodo del proyecto terminase, era una condición crucial.

3.2 Solución a Aplicar

La solución a aplicar sería la segunda descrita en el apartado anterior: un sistema existente que fuese adaptado a las necesidades de la entidad y que permitiese la ampliación por medio de nuevos módulos.

Con este objetivo en mente la solución debería partir de algún proyecto open source que hubiese demostrado ser un software de calidad y permitiese cubrir los requisitos bien porque ya los ofreciese o porque se pudiesen añadir ampliando la funcionalidad de la aplicación.

En los últimos años, han aparecido compañías que han hecho de la personalización de proyectos open source su modelo de negocio. Centrándose en el desarrollo personalizado y el soporte técnico más que en la venta de un producto de software cerrado.

Un proyecto interesante que actualmente permite este tipo de negocios ha sido OpenBravo. Un ERP libre que se distribuye bajo dos formatos (libre o licenciado con soporte técnico), que permite a terceras empresas construir su modelo de negocio sobre él.

Y es este sistema el que podría ser la base de la solución a aplicar en la entidad. Un software de gestión integral que en su implementación básica ya incluye muchos módulos de gestión económica, clientes, almacenes y proyectos.

Y si no fuera poco esto además posee cuatro características que la hacen ideal para escogerlo como base para llevar a cabo este proyecto:

- Se trata de una aplicación web, lo que permitirá su utilización desde las diferentes sedes repartidas por toda la península.
- Muchas empresas españolas proporcionan soporte a esta plataforma, con lo que una vez terminado el proyecto, está garantizado que pueda adaptarse a los cambios venideros.
- La arquitectura en la que se basa es MVC (modelo-vista-controlador) y el lenguaje sobre el que está construido es Java. Ambas características aseguran



un buen diseño y disponibilidad además de ser una arquitectura y un lenguaje familiares para el autor del proyecto.

- El proyecto es también siendo gestionado de una forma ejemplar, tanto por la documentación que ofrece a desarrolladores y usuarios, como el proceso de desarrollo y gestión de calidad que ha instaurado.

4 La herramienta

En este apartado se describe más a fondo el ERP Libre OpenBravo (3).

4.1 Características

El sistema Openbravo ERP se construye sobre la aplicación Openbravo, que proporciona un entorno de desarrollo apto para producir soluciones independientes del ERP o que lo complementen. Evidentemente el sistema ERP es una solución muy interesante que debería mantenerse abiertos a necesidades futuras, pero sin dejar que una demasiada funcionalidad oculte la que realmente es necesaria.

Esta herramienta como otros ERP, tratan de definir una capa de desarrollo que abstraiga la complejidad del sistema y permita centrarse en los modelos de negocio, dejando la seguridad, gestión de usuarios, comunicación y otras operaciones comunes al sistema.

- Desarrollo basado en modelos.
- Control de niveles de usuario por distribución organizativa y por roles.
- Arquitectura bien definida y escalable.

Openbravo ERP es un sistema OpenSource desarrollado por una empresa Navarra (Openbravo) que basa su modelo de negocio en ofrecer soporte a su red de Partners y formación a diferentes sectores.

Actualmente se encuentra entre uno de los proyectos más activos en Sourceforge (repositorio de proyectos opensource). Entre otras cosas por la gestión ejemplar que están haciendo del proyecto.

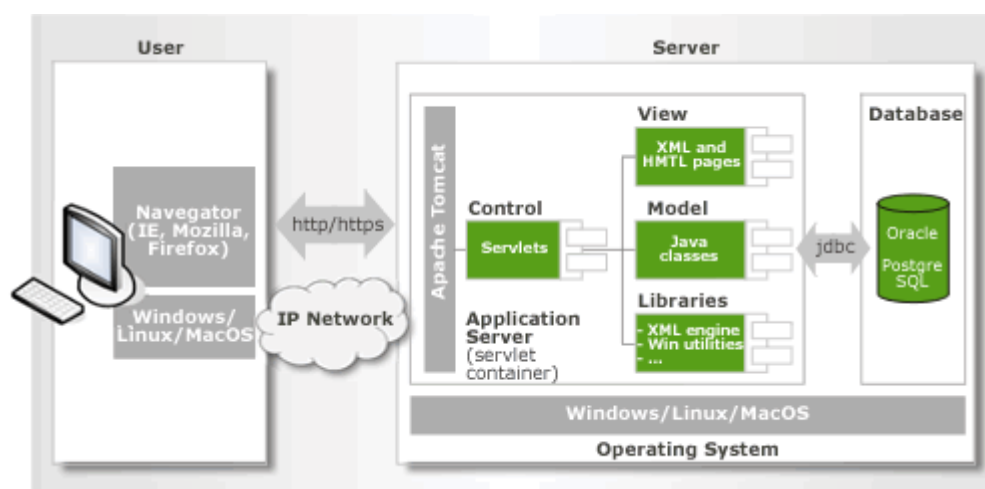


Ilustración 3: Despliegue Openbravo

A través de su página web coordinan y planifican las diferentes versiones de su producto. Además hacen un seguimiento muy cuidadoso de la calidad del producto mediante la automatización de pruebas.

4.2 Arquitectura

Openbravo es una plataforma que permite construir aplicaciones web y por lo tanto ha sido implementada definiendo una arquitectura típica de estas aplicaciones, arquitectura Modelo-Vista-Controlador (MVC en adelante).

La ventaja de esta arquitectura es que ha demostrado ser muy eficaz en cuanto al mantenimiento del código, por estar claramente separada la lógica de negocio de la presentación y los datos. Igualmente esta separación permite distribuir la aplicación en varios sistemas, construyendo sistemas escalables.

Actualmente hay varias tecnologías que implementan la arquitectura MVC y que permite construir aplicaciones escalables y distribuidas. Entre ellas además podemos encontrar varias en java, como Struts(4) o Spring. Openbravo en esencia constituye una nueva, ya que inicialmente no aprovechaba ninguna de estas tecnologías sino que las implementaba de forma propio. En la próxima versión de Openbravo (versión 2.50) la arquitectura cambiará para aprovecharse de los avances que ofrecen estas tecnologías, en concreto se construye sobre Spring(5) e Hibernate (6).

La arquitectura en Openbravo es crucial para entender cómo se pueden crear nuevos componentes o extender la funcionalidad existente.

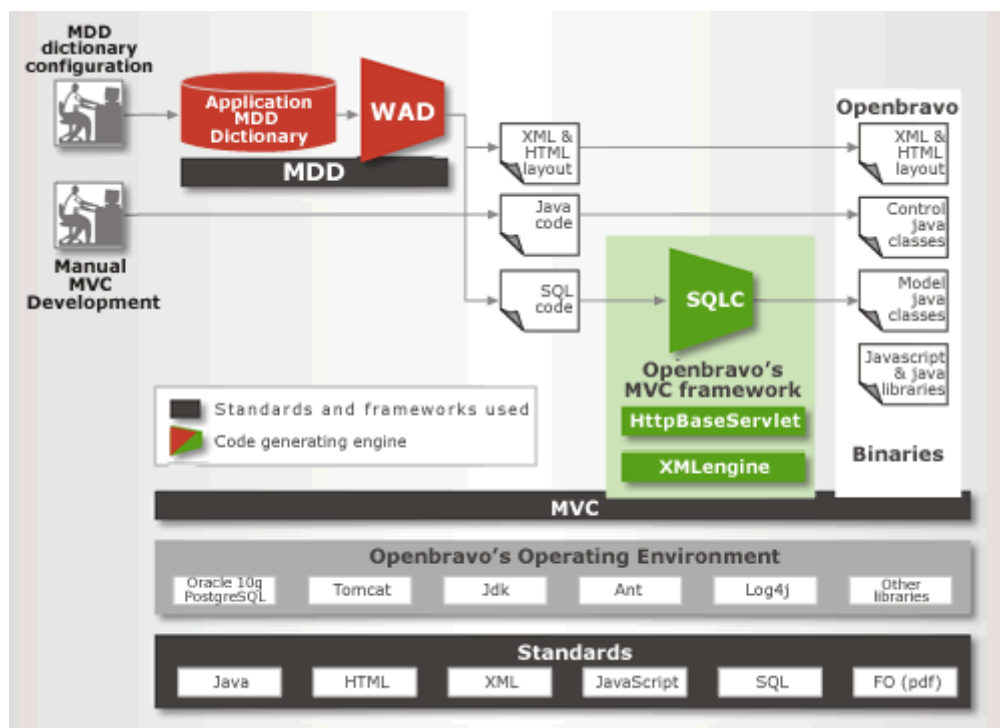


Ilustración 4: Arquitectura Openbravo

Las aplicaciones se construyen inicialmente definiendo el modelo de datos que tratará (las tablas de la base de datos). Estas tablas deben ser registradas en la aplicación para poder acceder a ellas a través de los componentes de una forma transparente.

En segundo lugar es necesario definir la vista del componente. Es decir todos los formularios y campos que permitirán acceder a la información. Este proceso es muy intuitivo y resulta bastante completo para la mayoría de las aplicaciones. Si alguna funcionalidad no existe es necesario crearla de cero. Como campos de acceso a gráficos, tablas especiales, etc. El proceso de crear un nuevo elemento no es sencillo y requiere de un conocimiento profundo del sistema, ya que es necesario programar el acceso que hace este elemento al modelo, la lógica de negocio, y las vistas que implementa.

Alternativamente también es posible definir informes. Es un proceso más manual, aunque es posible usar algunas herramientas de soporte como iReport(7).

En tercer lugar es posible cambiar la lógica de negocio que afecta a los componentes y tratar los datos como sea necesario. Este es el proceso más manual que implica programación. La lógica por defecto es leer y escribir las entradas de las tablas definidas en el primer paso.

A partir de los elementos definidos, Openbravo compila los nuevos elementos y los integra como parte de su sistema. La ventaja de este proceso de compilación es que frente a otras tecnologías que resultan muy pesadas por todas las capas que implementan, en el caso de Openbravo, se genera código autosuficiente (vista, controlador y acceso al modelo de datos, se traduce en una sola clase).

Esto requiere sin embargo, que por cada modificación que se haga en el modelo, vista o controlador de la aplicación, volver a compilar los elementos afectados y reempaquetar todo el sistema.

Openbravo también fue diseñado para soportar la multi-localización, lo que permite tener el sistema traducido a varios idiomas.

4.3 Soporte

Una de las características que han permitido seleccionar Openbravo como la herramienta sobre la que se construiría el proyecto es la existencia de varias empresas que ofrecen soporte técnico y formativo para esta plataforma.

En la página oficial del proyecto ofrecen una relación de empresas que ofrecen diferentes niveles de soporte. Esta red de Partners se distribuye por todo el territorio nacional y tiene presencia en algunos de los lugares más relevantes para la asociación, como Zaragoza y Madrid.

4.4 Documentación

La web oficial de l proyecto O penbravo E RP, o frece gr an ca ntidad de i nformación relacionada co n l a funcionalidad de l a aplicación, s u ar quitectura y guías par a co menzar a implementarla e i nstalarla (8). Sin embargo, no of rece una i nformación muy detallada, ya que esto se lo reserva a Partners y a sus cursos formativos.

Para complementar la base funcional que proporciona la aplicación y aspectos relacionados co n l a i nstalación adm inistración del s istema el li bro “*OPENBRAVO Guía del Usuario 2.3x*” (9).

Además al gunos usuarios y partners del pr oducto, ofrecen i nformación a t ravés d e internet, y entre ellos hay uno que proporciona tutoriales y documentación sobre diferentes funcionalidades y componentes de la aplicación, la empresa se denomina Telepieza (10).

Finalmente, a la par que se desarrollaba este proyecto, otro proyecto dirigido también al t ercer s ector fue concebido, d irigido a i nstituciones que t rabajan par a per sonas co n discapacidad (11). Esta web ha e laborado documentación del sistema, que aunque di rigida a su implementación, resulta de interés sobre todo la sección de instalación y administración.

Análisis (Diseño arquitectónico)

El diseño arquitectónico del sistema se realizó en base a la recopilación de un conjunto básico de requisitos de usuario, el análisis de las herramientas existentes y la problemática de la asociación, y las posibilidades ofrecidas por la herramienta seleccionada como base de la arquitectura.

En los siguientes puntos se recogen los principales elementos que han influido en el diseño.

1 Requisitos de usuario

En esta sección se recogen los requisitos de usuario fruto de las reuniones mantenidas con la gerencia de la asociación, el análisis de los procesos organizativos y la valoración de las posibles soluciones disponibles.

1.1 Requisitos Funcionales

ID: RF.1	Gestión de Participantes, Voluntarios, Trabajadores				
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta
Descripción					
Será necesario poder almacenar la información de contacto de los participantes, voluntarios y trabajadores. Incluyendo información como: Nombre, Apellidos, Fecha de Nacimiento, Dirección, Correo Electrónico, Teléfono y Nombre padre ó tutor.					

Req. Usuario 1: RF.1 - Gestión de Participantes, Voluntarios, Trabajadores

ID: RF.1.1	Incluir fotografía de participante				
Importancia	Deseable	Estado	Aceptado	Estabilidad	Alta
Descripción					
El sistema deberá proporcionar un mecanismo para incluir una fotografía del usuario, bien por medio de un archivo o de la captura por medio de una webcam.					

Req. Usuario 2: RF.1.1 - Incluir fotografía de participante

ID: RF.1.2	Información académica				
Importancia	Deseable	Estado	Aceptado	Estabilidad	Alta
Descripción					
El sistema deberá almacenar la información académica, cualidades, experiencia, preferencias y disponibilidad (de cara al voluntariado).					

Req. Usuario 3: RF.1.2 - Información académica

ID: RF.1.3	Comunicaciones a petición				
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta
Descripción					
En la información del participante, se deberá indicar si dicho participante está dispuesto a recibir comunicaciones vía e-mail, correo postal o SMS.					

Req. Usuario 4: RF.1.3 - Comunicaciones a petición

ID: RF.1.4	Adjuntar archivos				
Importancia	Deseable	Estado	Aceptado	Estabilidad	Alta
Descripción					
Debe ser posible incluir en las fichas de los usuarios archivos adjuntos para así poder incluir fotocopias de la cartilla de la seguridad social, o documentos que se requieren en algunos casos.					

Req. Usuario 5: RF.1.4 - Adjuntar archivos

ID: RF.2	Gestión de Areas				
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta
Descripción					
Las Areas agrupan diferentes tipos de proyectos, deberá ser posible definir diferentes areas.					

Req. Usuario 6: RF.2 - Gestión de Areas

ID: RF.3	Gestión de proyectos				
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta
Descripción					
Los proyectos agrupan diferentes programas. Deberá poder definirse los proyectos, definiendo las fechas que cubre, la región donde se realiza y el principal responsable.					

Req. Usuario 7: RF.3 - Gestión de proyectos

ID: RF.3.1	Documentación adjunta al proyecto				
Importancia	Deseable	Estado	Aceptado	Estabilidad	Alta
Descripción					
La ficha del proyecto debe permitir adjuntar la documentación que se generó en el proyecto (documentos de propuesta, evaluaciones, etc.)					

Req. Usuario 8: RF.3.1 - Documentación adjunta al proyecto

ID: RF.4	Gestión de programas				
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta
Descripción					
Los proyectos se dividen en programas, que será la aplicación local de un determinado proyecto.					

Req. Usuario 9: RF.4 - Gestión de programas

ID: RF.4.1	Definición de nuevos parámetros				
Importancia	Media	Estado	Aceptado	Estabilidad	Alta
Descripción					
Los programas y actividades deberán permitir definir parámetros nuevos para incluir en las fichas de inscripción de los participantes. Esto parámetros podrán ser configurados por los responsables de dar de alta los programas o actividades.					

Req. Usuario 10: RF.4.1 - Definición de nuevos parámetros

ID: RF.5	Gestión de actividades				
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta
Descripción					
Los programas pueden contener además actividades puntuales. Puesto que una actividad forma parte de un programa determinado, el coste de dicha actividad podrá ser independiente o ser cero por considerarse incluido en el coste del programa.					

Req. Usuario 11: RF.5 - Gestión de actividades

ID: RF.6	Precio de socio				
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta
Descripción					
Tanto en la configuración del programa, como en la actividad, deberá ser posible indicar dos precios alternativos de actividad, uno para el participante normal y otro para el socio.					

Req. Usuario 12: RF.6 - Precio de socio

ID: RF.7	Inscripción de participantes				
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta
Descripción					
Una vez definido un programa y un participante, será posible inscribir al participante en dicho programa. Así mismo una vez inscrito en programa, el participante podrá ser inscrito en las actividades definidas en dicho programa.					

Req. Usuario 13: RF.7 - Inscripción de participantes

ID: RF.8	Imputación de gastos				
Importancia	Media	Estado	Rechazado	Estabilidad	Alta
Descripción					
El sistema deberá permitir definir los gastos que se han producido por la realización de una determinada actividad, programa o proyecto. Junto con el gasto se podrá incluir el justificante.					
Comentarios					
Se pospone temporalmente.					

Req. Usuario 14: RF.8 - Imputación de gastos

ID: RF.9	Elaboración de recibos				
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta
Descripción					
Al inscribir a un participante en un programa o actividad, el sistema deberá permitir generar un recibo de pago, con el concepto de pago de ese servicio.					

Req. Usuario 15: RF.9 - Elaboración de recibos

ID: RF.9.1	Numeración única de recibos				
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta
Descripción					
La numeración que identifique los recibos debe ser única y configurable.					

Req. Usuario 16: RF.9.1 - Numeración única de recibos

ID: RF.9.2	Recibos impagados				
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta
Descripción					
El sistema deberá permitir visualizar los recibos pendientes de pago.					

Req. Usuario 17: RF.9.2 - Recibos impagados

ID: RF.9.3	Generación automática de recibos				
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta
Descripción					
Algunos servicios pueden definirse como de pago mensual. En caso de que un participante no pague el mes correspondiente el sistema deberá permitir acceder a los recibos atrasados, para cumplir los pagos.					

Req. Usuario 18: RF.9.3 - Generación automática de recibos

ID: RF.9.4	Factura de servicio				
Importancia	Media	Estado	Rechazado	Estabilidad	Alta
Descripción					
A partir de un conjunto de recibos será posible generar una factura.					
Comentarios					
Se pospone temporalmente.					

Req. Usuario 19: RF.9.4 - Factura de servicio

ID: RF.10	Introducción de facturas de compra				
Importancia	Deseable	Estado	Rechazado	Estabilidad	Alta
Descripción					
Para agilizar el proceso de seguimiento contable de los programas, el sistema deberá permitir ingresar las facturas de gastos imputables a esos servicios.					
Comentarios					
Se pospone temporalmente.					

Req. Usuario 20: RF.10 - Introducción de facturas de compra

ID: RF.11	Gestión de sedes				
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta
Descripción					
La organización se divide geográficamente en diferentes sedes, que pueden o no compartir actividades. Pero que por lo general tendrán acceso a todos los participantes.					

Req. Usuario 21: RF.11 - Gestión de sedes

ID: RF.12	Gestión de usuarios				
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta
Descripción					
Los empleados de la organización deberán tener acceso a la aplicación, bajo diferentes permisos.					

Req. Usuario 22: RF.12 - Gestión de usuarios

ID: RF.13	Gestión de roles				
Importancia	Deseable	Estado	Aceptado	Estabilidad	Alta
Descripción					
Los permisos de los usuarios se gestionará por medio del concepto de roles. De esta forma varios usuarios podrán compartir el mismo tipo de acceso a través de un mismo rol.					
Comentarios					
Funcionalidad ofrecida por el sistema.					

Req. Usuario 23: RF.13 - Gestión de roles

ID: RF.14	Capacidad de comunicación por SMS				
Importancia	Deseable	Estado	Aceptado	Estabilidad	Alta
Descripción					
Uno de los procesos habituales de la asociación es la de informar a los destinatarios de ciertos eventos o servicios. Sería deseable contar con un método que permitiese mandar SMS a los destinatarios de un determinado programa.					

Req. Usuario 24: RF.14 - Capacidad de comunicación por SMS

ID: RF.L.1	Listado de participantes de programa				
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta
Descripción					
El sistema podrá generar listados de los participantes de un programa determinado, ordenando a los participantes por apellido y nombre.					

Req. Usuario 25: RF.L.1 - Listado de participantes de programa

ID: RF.L.2	Ficha resumen de programa				
Importancia	Deseable	Estado	Aceptado	Estabilidad	Alta
Descripción					
Se podrá generar un informe de un programa que incluya toda la información recogida en su ficha, junto a los ingresos y gastos, y el listado de participantes, con los atributos especiales definidos.					

Req. Usuario 26: RF.L.2 - Ficha resumen de programa

ID: RF.L.3	Listado de participantes de actividad				
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta
Descripción					
El sistema podrá generar listados de los participantes de una actividad determinada, ordenando a los participantes por apellido y nombre.					

Req. Usuario 27: RF.L.3 - Listado de participantes de actividad

ID: RF.L.4	Ficha resumen de actividad				
Importancia	Deseable	Estado	Aceptado	Estabilidad	Alta
Descripción					
Se podrá generar un informe de una actividad que incluya toda la información recogida en su ficha, junto a los ingresos y gastos, y el listado de participantes, con los atributos especiales definidos.					

Req. Usuario 28: RF.L.4 - Ficha resumen de actividad

ID: RF.L.5	Listados de proyectos por área				
Importancia	Deseable	Estado	Aceptado	Estabilidad	Alta
Descripción					
Se podrá generar un listado de proyectos por área de actividad.					

Req. Usuario 29: RF.L.5 - Listados de proyectos por área

ID: RF.L.6	Listados de programas por proyecto				
Importancia	Deseable	Estado	Aceptado	Estabilidad	Alta
Descripción					
Se podrá generar un listado de programas por proyecto.					

Req. Usuario 30: RF.L.6 - Listados de programas por proyecto

ID: RF.L.7	Informe de participación				
Importancia	Deseable	Estado	Aceptado	Estabilidad	Alta
Descripción					
Se podrá generar un informe de participación por proyecto. En la que recoja el número de participantes en los diferentes programas.					

Req. Usuario 31: RF.L.7 - Informe de participación

ID: RF.L.8	Listados de participantes por áreas de interés				
Importancia	Deseable	Estado	Aceptado	Estabilidad	Alta
Descripción					
Se podrá generar un listado de participantes por áreas de interés.					

Req. Usuario 32: RF.L.8 - Listados de participantes por áreas de interés

ID: RF.L.9	Listado de Voluntarios				
Importancia	Deseable	Estado	Aceptado	Estabilidad	Alta
Descripción					
El sistema permitirá generar un listado de los voluntarios de la asociación, ordenados por apellido y nombre.					

Req. Usuario 33: RF.L.9 - Listado de Voluntarios

ID: RF.L.10	Listados en formatos especiales, mailing				
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta
Descripción					
El sistema deberá permitir realizar impresiones de etiquetas de mailing, con la información de contacto postal de los participantes, trabajadores y/o colaboradores, a través de varios criterios de búsqueda.					

Req. Usuario 34: RF.L.10 - Listados en formatos especiales, mailing

ID: RF.L.11	Ficha resumen de participante				
Importancia	Deseable	Estado	Aceptado	Estabilidad	Alta
Descripción					
El sistema permitirá imprimir un informe del cada participante, en el que se recoja los datos de contacto, su fotografía, y el historial de participación en la asociación (inscripciones en actividades y programas).					

Req. Usuario 35: RF.L.11 - Ficha resumen de participante

ID: RF.L.12	Listado de mailing				
Importancia	Deseable	Estado	Aceptado	Estabilidad	Alta
Descripción					
Se podrá generar un listado en excel con las direcciones de participantes, mediante diferentes filtros. Esta característica es necesaria para el envío de publicidad postal a través de terceras empresas.					

Req. Usuario 36: RF.L.12 - Listado de mailing

1.2 Requisitos No Funcionales

ID: RNF.1	Acceso simultaneo						
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta	Tipo	Disponibilidad
Descripción							
Varios usuario accederán simultaneamente a la aplicación, y es necesario que existan consistencia en todo momento, de la información.							

Req. No Funcional 1: RNF.1 - Acceso simultaneo

ID: RNF.2	Acceso distribuído						
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta	Tipo	Disponibilidad
Descripción							
La organización se distribuye en diferentes regiones geográficas, es necesario sin embargo, que todos puedan acceder igualmente al aplicativo.							

Req. No Funcional 2: RNF.2 - Acceso distribuído

ID: RNF.3	Multisistema						
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta	Tipo	Accesibilidad
Descripción							
El sistema debe funcionar bajo diferentes sistemas operativos, ya que se prevee en corto periodo de tiempo migrar algunos sistemas a GNU/Linux.							

Req. No Funcional 3: RNF.3 – Multisistema

ID: RNF.4	Disponibilidad durante jornada laborable						
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta	Tipo	Disponibilidad
Descripción							
El sistema debe estar disponible durante las jornadas laborables y ocasionalmente en fin de semana, desde las 9 de la mañana a las 8 de la tarde.							

Req. No Funcional 4: RNF.4 - Disponibilidad durante jornada laborable

ID: RNF.5	Sistema intuitivo y autoformativo						
Importancia	Deseable	Estado	Aceptado	Estabilidad	Alta	Tipo	Interfaz
Descripción							
El perfil de los usuarios potenciales es poco técnico, así el sistema deberá de ser lo más intuitivo posible.							

Req. No Funcional 5: RNF.5 - Sistema intuitivo y autoformativo

ID: RNF.6	Ayuda disponible						
Importancia	Baja	Estado	Aceptado	Estabilidad	Alta	Tipo	Interfaz
Descripción							
La aplicación deberá proporcionar medios formativos, previendo la rotación futura de personal.							

Req. No Funcional 6: RNF.6 - Ayuda disponible

ID: RNF.7	Garantizar autenticación						
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta	Tipo	Seguridad
Descripción							
Es necesario asegurar que el acceso a la aplicación solo se conceda a personas autorizadas.							

Req. No Funcional 7: RNF.7 - Garantizar autenticación

ID: RNF.8	Garantizar la mantenibilidad del software						
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta	Tipo	Diseño
Descripción							
El importante que el software generado pueda ser transferido de un proveedor de servicios a otro, para garantizar la continuidad del software incluso si una vez terminado este proyecto el autor desaparece.							

Req. No Funcional 8: RNF.8 - Garantizar la mantenibilidad del software

ID: RNF.9	Sistema multiidioma						
Importancia	Deseable	Estado	Aceptado	Estabilidad	Alta	Tipo	Accesibilidad
Descripción							
En coherencia con la distribución geográfica, el sistema debería estar disponible en diferentes idiomas.							

Req. No Funcional 9: RNF.9 - Sistema multiidioma

ID: RNF.10	Backup automatizado						
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta	Tipo	Seguridad
Descripción							
Un sistema de backup automatizado deberá estar disponible para que no sea necesario una asistencia manual, y que garantice la no pérdida de datos.							

Req. No Funcional 10: RNF.10 - Backup automatizado

ID: RNF.11	Capaz de funcionar con los recursos disponibles						
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta	Tipo	Rendimiento
Descripción							
La organización dispone de algunos sistemas informáticos que habrá que utilizar minimizando al máximo la adquisición de nuevos equipos o servicios externos.							

Req. No Funcional 11: RNF.11 - Capaz de funcionar con los recursos disponibles

ID: RNF.12	Migración de datos						
Importancia	Alta	Estado	Aceptado	Estabilidad	Alta	Tipo	Disponibilidad
Descripción							
Se deberán migrar los datos existentes de las aplicaciones actuales, para evitar la insercción manual en la nueva aplicación.							

Req. No Funcional 12: RNF.12 - Migración de datos

ID: RNF.13	Independencia del servidor de SMS						
Importancia	Deseable	Estado	Aceptado	Estabilidad	Alta	Tipo	Diseño
Descripción							
La asociación cuenta con un proveedor de envío de SMS, que es Arsys. El sistema deberá funcionar con este proveedor. Sin embargo se desea que el sistema pueda funcionar igualmente con diferentes servidores y cuentas.							

Req. No Funcional 13: RNF.13 - Independencia del servidor de SMS

ID: RNF.14	Minimizar costes de comunicación						
Importancia	Deseable	Estado	Aceptado	Estabilidad	Alta	Tipo	Rendimiento
Descripción							
Al realizar los envíos por SMS, o los listados de mailing se tendrá cuidado de no repetir número de teléfono o dirección. Unificando en un mismo envío personas que compartan residencia.							

Req. No Funcional 14: RNF.14 - Minimizar costes de comunicación

2 Modelo conceptual

En la Ilustración 5 se muestra el modelo conceptual de los principales elementos a tener en cuenta del sistema. En él se puede visualizar la relación de dependencia que existe entre los diferentes elementos, y cuál es la principal diferencia entre la inscripción en un programa y una actividad.

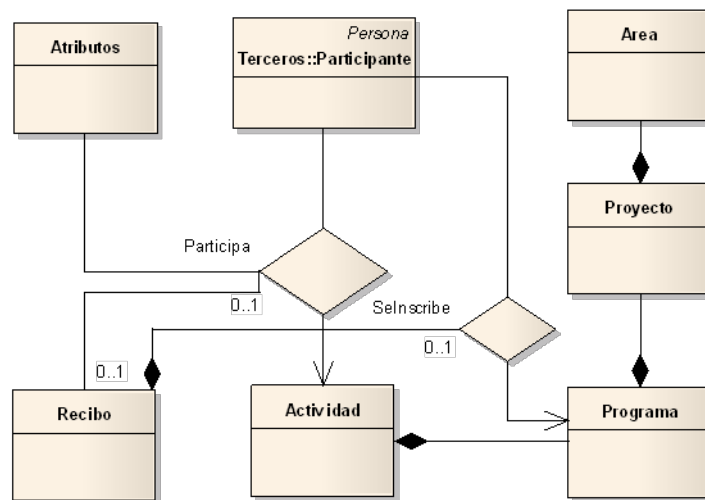


Ilustración 5: Modelo conceptual de los elementos claves del sistema

En la Ilustración 6 se muestra el tipo de personas que gestionaría la aplicación. El usuario de la aplicación corresponderá con el rol Trabajador.

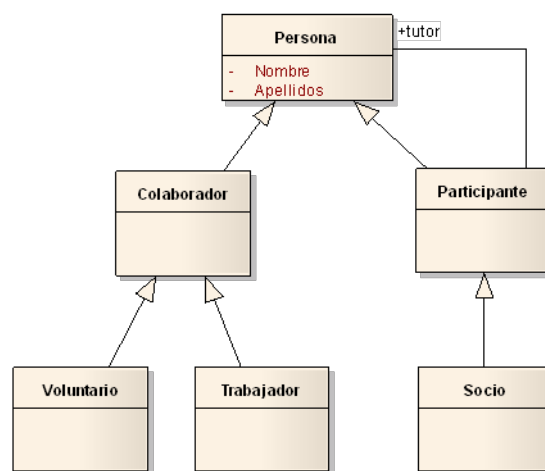


Ilustración 6: Modelo conceptual tipos de persona

3 Arquitectura del sistema

El sistema Openbravo establece una base sólida sobre la que está construido el sistema ERP, y que ofrece una gran cantidad de funcionalidad base de la que no hay que preocuparse durante el diseño del sistema, sino que se abordará más adelante durante la configuración.

En la Ilustración 7 se puede ver cómo el elemento ERP de Openbravo se sustenta sobre los servicios de gestión de usuarios y acceso al modelo de datos, como lo hará también los módulos que se desarrollen para cubrir los requisitos.

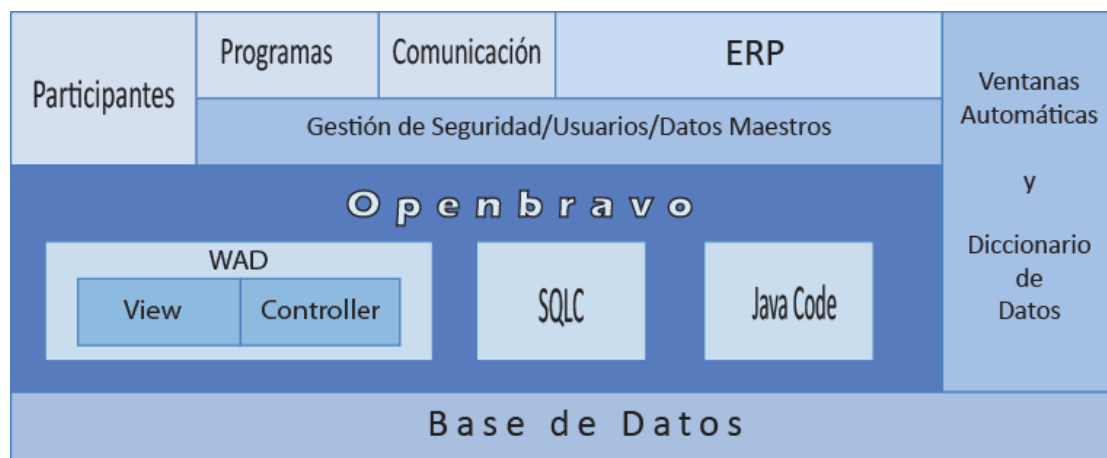


Ilustración 7: Arquitectura YMCA Openbravo

Algunos de los elementos de la plataforma se reutilizarán, otros no y algunos habrá que realizarlos desde cero o a partir de otros existentes. Así se hace necesario estudiar los módulos existentes y el tipo de funcionalidad que ofrecen para ver en qué grado pueden ser reutilizados.

Funcionalidades de la aplicación Openbravo ERP:

Tabla 1: Funcionalidad existente en Openbravo ERP

Funcionalidad	Módulo	Se mantiene	Resuelve
Gestión de terceros	Datos maestros	Sí	Gestión de Participantes (Parcialmente)
Gestión de productos	Datos maestros	No	Algunas características de la gestión de productos que se encuentran en este módulo se reutilizarán.
Gestión de usuarios y roles	Configuración general	Sí	Toda la gestión de la seguridad de acceso de la aplicación.
Gestión de usuarios	Gestión de datos Maestros	Sí	Es la base del control de acceso de la aplicación. Además la información de contacto se mantendrá en este módulo.
Gestión de compras	-	No	
Gestión de almacén	-	No	Este módulo puede ser utilizado en el futuro para controlar el inventario de material de cada sede.
Gestión de producción	-	No	
Gestión MRP	-	No	
Gestión de Ventas	-	No	
Gestión de Proyectos y Servicios	-	No	La gestión de proyectos está muy alejada de la necesidad que existe en la asociación, el módulo de Openbravo ERP va más enfocado a la gestión de Proyectos de construcción.
Gestión financiera	-	No	Se aplaza esta funcionalidad
Diccionario de la aplicación	-	Sí	Es un elemento clave de la arquitectura, permite generar código automático a partir de la definición de las tablas de datos y las ventanas de edición de estos datos.

A parte de los módulos que se reutilicen hay que integrar en el sistema los siguientes:

- Gestión de Áreas
- Gestión de Proyectos
- Gestión de Programas y actividades
- Gestión de Participantes
- Gestión de inscripciones
- Generación de Recibos
- Servicio de comunicación

Estos módulos aparecen parcialmente reflejados en la Ilustración 7, y se integraría al mismo nivel dentro de la aplicación.

En el siguiente capítulo se definen en detalle cada uno de estos módulos.

4 Requisitos de Software

De análisis de los requisitos de usuario, las restricciones impuestas por los requisitos funcionales y el estudio de la arquitectura Openbravo, se han extraído los siguientes requisitos de software, que definirán el diseño detallado del sistema.

ID: SR.1	Alta de Participantes, Voluntarios, Trabajadores				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.1
Descripción					
<p>Será necesario poder almacenar la información de contacto de los participantes, voluntarios y trabajadores. La información almacenada será la siguiente:</p> <ul style="list-style-type: none"> • Sede desde donde fue dado de alta • Nombre • Apellidos • Fecha de Nacimiento • DNI • Dirección postal (con Calle, Municipio, Provincia y País) • Correo Electrónico • Teléfono móvil • Teléfono alternativo • Fax • Nombre padre ó tutor • Idioma • Descripción • Comentarios 					

Req. Software 1: SR.1 - Alta de Participantes, Voluntarios, Trabajadores

ID: SR.2	Incluir fotografía de participante				
Importancia	Deseable	Estado	Aceptado	Dependencia	RF.1.1
Descripción					
<p>El sistema deberá proporcionar un mecanismo para incluir una fotografía del usuario, bien por medio de un archivo o de la captura por medio de una webcam. La imagen tendrá una dimensión máxima de 150x150 pixels, y será redimensionada en caso de que supere estas dimensiones.</p>					

Req. Software 2: SR.2 - Incluir fotografía de participante

ID: SR.3	Información académica				
Importancia	Deseable	Estado	Aceptado	Dependencia	RF.1.2
Descripción					
<p>El sistema deberá almacenar información de interés acerca del perfil del voluntario/colaborador. Incluyendo la siguiente información:</p> <ul style="list-style-type: none"> • Profesión • Estudios • Experiencia • Disponibilidad • Titulación de monitor y/o coordinador: campos tipo checkbox • Comentarios 					

Req. Software 3: SR.3 - Información académica

ID: SR.4	Comunicaciones a petición				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.1.3
Descripción					
<p>En la información del participante, se deberá indicar si dicho participante está dispuesto a recibir comunicaciones vía e-mail, correo postal o SMS. Por defecto todo usuario se configura como que acepta todo tipo de comunicación.</p>					

Req. Software 4: SR.4 - Comunicaciones a petición

ID: SR.5	Adjuntar archivos				
Importancia	Deseable	Estado	Aceptado	Dependencia	RF.1.4
Descripción					
<p>Debe ser posible incluir en las fichas de los usuarios archivos adjuntos para así poder incluir fotocopias de la cartilla de la seguridad social, o documentos que se requieren en algunos casos. Cada archivo incluirá una definición del tipo de archivo y una descripción.</p>					

Req. Software 5: SR.5 - Adjuntar archivos

ID: SR.6	Baja de participantes				
Importancia	Deseable	Estado	Aceptado	Dependencia	RF.1
Descripción					
<p>Un registro de participante se podrá eliminar únicamente del sistema si no ha tenido ninguna relación con la entidad. Lo que significa que no se ha inscrito a ningún programa o actividad. En caso de cesar la relación con el participante, y evitar que le lleguen comunicados el sistema permitirá desactivar al usuario.</p>					

Req. Software 6: SR.6 - Baja de participantes

ID: SR.7	Gestión de Areas				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.2, RNF.15
Descripción					
<p>Las Areas agrupan diferentes tipos de proyectos, deberá ser posible definir diferentes areas.</p>					

Req. Software 7: SR.7 - Gestión de Areas

ID: SR.8	Alta de proyectos				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.3, RNF.15
Descripción					
<p>Los proyectos agrupan diferentes programas. El sistema permitirá dar de alta nuevos proyectos con la siguiente información:</p> <ul style="list-style-type: none"> • Área a la pertenece • Identificador: Valor único. • Nombre • Descripción • Observaciones • Responsable: Referencia a un trabajador de la entidad. • Fecha de inicio • Fecha de finalización 					

Req. Software 8: SR.8 - Alta de proyectos

ID: SR.9	Documentación adjunta al proyecto				
Importancia	Deseable	Estado	Aceptado	Dependencia	RF.3.1
Descripción					
<p>La ficha del proyecto debe permitir adjuntar la documentación que se generó en el proyecto (documentos de propuesta, evaluaciones, etc.) Cada archivo incluirá una definición del tipo de archivo y una descripción.</p>					

Req. Software 9: SR.9 - Documentación adjunta al proyecto

ID: SR.10	Baja de proyectos				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.3
Descripción					
<p>Un proyecto sólo se podrá eliminar si se eliminan todos los programas asociados al proyecto.</p>					

Req. Software 10: SR.10 - Baja de proyectos

ID: SR.11	Alta de programas			
Importancia	Alta	Estado	Aceptado	Dependencia RF.4, RNF.15
Descripción				
<p>Los proyectos se dividen en programas, que será la aplicación local de un determinado proyecto. Un programa deberá contener la siguiente información:</p> <ul style="list-style-type: none"> • Sede que realiza el programa • Proyecto al que pertenece • Identificador: Valor único. • Nombre • Descripción • Fecha de inicio • Fecha de finalización • Conjunto de atributos: Referencia a un conjunto predefinido. • Observaciones • Responsable • Coste Normal • Coste Socio • Pago mensual • Gastos • Ingresos: Campo automático, generado a partir de las inscripciones. 				
Req. Software 11: SR.11 - Alta de programas				

ID: SR.12	Baja de programa			
Importancia	Alta	Estado	Aceptado	Dependencia RF.4
Descripción				
Un programa puede ser eliminado si no tiene inscripciones, ni actividades asociadas.				
Req. Software 12: SR.12 - Baja de programa				

ID: SR.13	Definición de atributos			
Importancia	Media	Estado	Aceptado	Dependencia RF.4.1
Descripción				
<p>Los programas y actividades deberán permitir definir parámetros nuevos para incluir en las fichas de inscripción de los participantes. Estos parámetros podrán constar fundamentalmente de diferentes atributos. Cada atributo tendrá la siguiente información:</p> <ul style="list-style-type: none"> • Nombre • Descripción • Tipo • Obligatoriedad: Indica si ese campo será obligatorio rellenarlo. <p>El tipo de atributo podrá ser una lista, en cuyo caso se podrán indicar los valores de la lista.</p>				
Req. Software 13: SR.13 - Definición de atributos				

ID: SR.14	Definición de conjuntos de atributos				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.4.1
Descripción					
<p>Los programas y actividades deberán permitir definir parámetros nuevos para incluir en las fichas de inscripción de los participantes. Al definir el programa, o actividad, se seleccionará conjunto de atributos. Este conjunto de atributos debe estar preconfigurado con la siguiente información:</p> <ul style="list-style-type: none"> • Nombre de atributo • Descripción • Listado de atributos: seleccionar tantos atributos predefinidos como se desee. 					

Req. Software 14: SR.14 - Definición de conjuntos de atributos

ID: SR.15	Alta de actividades				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.5, RNF.15
Descripción					
<p>Los programas pueden contener actividades. Para poder dar de alta una actividad es necesario que exista un programa al que pertenezca esta actividad. La información que contendrá la actividad será:</p> <ul style="list-style-type: none"> • Sede que realiza la actividad • Programa al que pertenece • Identificador • Nombre • Descripción • Fecha de inicio • Fecha de finalización • Capacidad • Conjunto de atributos • Observaciones • Responsable • Coste incluido: En caso de que el coste esté incluido en la programa, no será necesario completar los dos campos siguientes. • Coste normal • Coste de socio • Ingresos: Campo automático, generado a partir de las inscripciones. • Gastos 					

Req. Software 15: SR.15 - Alta de actividades

ID: SR.16	Baja de actividades				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.5
Descripción					
<p>Para poder dar de baja una actividad del sistema es necesario que no exista ninguna inscripción realizada a esta actividad.</p>					

Req. Software 16: SR.16 - Baja de actividades

ID: SR.17	Precio de socio				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.6
Descripción					
<p>Tanto en la configuración del programa, como en la actividad, deberá ser posible indicar dos precios alternativos de actividad, uno para el participante normal y otro para el socio. Esta tarifa se establece en la moneda local, en este caso euros.</p>					

Req. Software 17: SR.17 - Precio de socio

ID: SR.18	Inscripción de participantes				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.7
Descripción					
<p>Una vez definido un programa y un participante, será posible inscribir al participante en dicho programa. Así mismo una vez inscrito en programa, el participante podrá ser inscrito en las actividades definidas en dicho programa. La inscripción incluirá:</p> <ul style="list-style-type: none"> • La fecha de inicio de participación en el programa o actividad • El coste: el coste corresponde al de la actividad o programa, pero puede ser editado. • El valor de los parámetros personalizables • Observaciones • El total de recibos impagados: Campo automático. 					

Req. Software 18: SR.18 - Inscripción de participantes

ID: SR.19	Cancelación de inscripción				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.7
Descripción					
<p>Una inscripción puede cancelarse mientras que no se realice un recibo de pago. En cualquier caso se considera finaliza la inscripción si se desactiva la inscripción.</p>					

Req. Software 19: SR.19 - Cancelación de inscripción

ID: SR.20	Elaboración de recibos				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.9
Descripción					
<p>Al inscribir a un participante en un programa o actividad, el sistema deberá permitir generar un recibo de pago, con el concepto de pago de ese servicio.</p>					

Req. Software 20: SR.20 - Elaboración de recibos

ID: SR.21	Numeración única de recibos				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.9.1
Descripción					
<p>La numeración que identifique los recibos debe ser única y configurable.</p>					

Req. Software 21: SR.21 - Numeración única de recibos

ID: SR.22	Recibos impagados				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.9.2
Descripción					
El sistema deberá permitir visualizar los recibos pendientes de pago.					

Req. Software 22: SR.22 - Recibos impagados

ID: SR.23	Generación automática de recibos				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.9.3
Descripción					
Algunos servicios pueden definirse como de pago mensual. En caso de que un participante no pague el mes correspondiente el sistema deberá permitir acceder a los recibos atrasados, para cumplir los pagos.					

Req. Software 23: SR.23 - Generación automática de recibos

ID: SR.24	Gestión de sedes				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.11
Descripción					
<p>La organización se divide geográficamente en diferentes sedes, que pueden o no compartir actividades. Pero que por lo general tendrán acceso a todos los participantes. Cada sede se dará de alta con la siguiente información:</p> <ul style="list-style-type: none"> • Identificador • Nombre • Descripción • Dirección • Teléfono • Teléfono Alternativo • Fax • Correo electrónico • NIF • Responsable 					

Req. Software 24: SR.24 - Gestión de sedes

ID: SR.25	Alta de usuarios				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.12
Descripción					
<p>Los empleados de la organización deberán tener acceso a la aplicación, bajo diferentes permisos. Al realizar el alta de un usuario se partirá de un participante creado y que pertenezca al grupo de trabajadores, además se incluirá la siguiente información:</p> <ul style="list-style-type: none"> • Nombre usuario: debe ser único en el sistema. • Contraseña • Listado de roles de acceso 					
Comentarios					
Característica del sistema Openbravo.					

Req. Software 25: SR.25 - Alta de usuarios

ID: SR.26	Baja de usuarios				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.12
Descripción					
<p>Los usuarios, como participantes del sistema, tienen las mismas limitaciones que los participantes para ser eliminados, sin embargo se podrá desactivar su acceso al sistema eliminando la contraseña de acceso, o los roles de acceso.</p>					

Req. Software 26: SR.26 - Baja de usuarios

ID: SR.27	Alta de roles				
Importancia	Deseable	Estado	Aceptado	Dependencia	RF.13, RNF.7
Descripción					
<p>Los permisos de los usuarios se gestionará por medio del concepto de roles. De esta forma varios usuarios podrán compartir el mismo tipo de acceso a través de un mismo rol. Cada rol contendrá la siguiente información:</p> <ul style="list-style-type: none"> • Nombre • Nivel de usuario • Acceso a Ventanas • Acceso a Procesos • Acceso a formularios • Acceso a tareas • Permisos a organizaciones: Organizaciones a las que tendrá acceso el rol. 					
Comentarios					
Característica del sistema Openbravo.					

Req. Software 27: SR.27 - Alta de roles

ID: SR.28	Baja de rol				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.13
Descripción					
<p>Los roles no asignados podrán ser eliminados, también pueden desactivarse impidiendo el acceso al sistema por medio de ese rol.</p>					

Req. Software 28: SR.28 - Baja de rol

ID: SR.29	Login en el sistema				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.13, RF.12, RNF.7
Descripción					
<p>Al entrar un usuario en el sistema se requerirá su nombre de usuario y contraseña, y accederá con el rol configurado por defecto. Si este está desactivado accederá por el primero activo.</p>					

Req. Software 29: SR.29 - Login en el sistema

ID: SR.30	Cambio de rol				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.13
Descripción					
<p>Un usuario activo del sistema, puede seleccionar con que rol actuar de todos los que tenga asignados.</p>					

Req. Software 30: SR.30 - Cambio de rol

ID: SR.31	Alta de envío de SMS				
Importancia	Deseable	Estado	Aceptado	Dependencia	RF.14
Descripción					
<p>El sistema permitirá enviar un SMS a un conjunto de personas indicando: Cuenta de envío del SMS, Remitente, Mensaje (160 caracteres). Y un filtro a partir de:</p> <ul style="list-style-type: none">• Áreas de interés• Grupo de terceros• Área de actividad• Proyecto• Programa• Actividad• Tipo de usuario					

Req. Software 31: SR.31 - Alta de envío de SMS

ID: SR.32	Alta manual de destinatarios SMS				
Importancia	Deseable	Estado	Aceptado	Dependencia	RF.14
Descripción					
<p>A parte de seleccionar usuarios como destinatarios con un filtro también será posible incluir los destinatarios manualmente.</p>					

Req. Software 32: SR.32 - Alta manual de destinatarios SMS

ID: SR.33	Baja de destinatarios SMS				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.14
Descripción					
<p>Antes de proceder al envío se podrá eliminar usuario destinatarios que hayan sido añadidos manualmente o por el filtro de envío.</p>					

Req. Software 33: SR.33 - Baja de destinatarios SMS

ID: SR.34	Alta de cuentas de envío de SMS				
Importancia	Deseable	Estado	Aceptado	Dependencia	RF.14
Descripción					
<p>Será posible configurar diferentes cuentas de envío de SMS indicando los siguientes datos:</p> <ul style="list-style-type: none">• Servicio de envío de SMS• Nombre• Descripción• Nombre de usuario• Contraseña					

Req. Software 34: SR.34 - Alta de cuentas de envío de SMS

ID: SR.35	Alta de nuevos servicios de envío de SMS				
Importancia	Deseable	Estado	Aceptado	Dependencia	RF.14, RNF.13
Descripción					
<p>Por medio de la implementación de la clase SMSGateway es posible añadir nuevos servicios de envío de SMS, la interfaz permite darlos de alta indicando la clase capaz de comunicarse con el servicio de envío de SMS.</p>					

Req. Software 35: SR.35 - Alta de nuevos servicios de envío de SMS

ID: SR.36	Unificación de destinatarios				
Importancia	Deseable	Estado	Aceptado	Dependencia	RNF.14
Descripción					
<p>Al aplicar el filtro de envío de SMS, si dos o más participantes comparten el mismo número de teléfono, se generará una única orden de envío para ese número.</p>					

Req. Software 36: SR.36 - Unificación de destinatarios

ID: SR.37	Historial de envío de SMS				
Importancia	Alta	Estado	Aceptado	Dependencia	RNF.14
Descripción					
<p>El sistema almacena el historial de envío de todos los SMS, junto con la respuesta proporcionada por el proveedor del servicio de envío de SMS.</p>					

Req. Software 37: SR.37 - Historial de envío de SMS

ID: SR.38	Listado de participantes de programa				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.L.1
Descripción					
<p>El sistema podrá generar listados de los participantes de un programa determinado, ordenando a los participantes por apellido y nombre. La información recogida en el listado será:</p> <ul style="list-style-type: none"> • Apellidos • Nombre • Teléfono • Edad • Fecha de Nacimiento • Teléfono alternativo • Correo electrónico 					

Req. Software 38: SR.38 - Listado de participantes de programa

ID: SR.39	Ficha resumen de programa				
Importancia	Deseable	Estado	Aceptado	Dependencia	RF.L.2
Descripción					
<p>Se podrá generar un informe de un programa que incluya toda la información recogida en su ficha, junto a los ingresos y gastos, y el listado de participantes, con los atributos especiales definidos.</p>					

Req. Software 39: SR.39 - Ficha resumen de programa

ID: SR.40	Listado de participantes de actividad				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.L.3
Descripción					
<p>El sistema podrá generar listados de los participantes de una actividad determinada, ordenando a los participantes por apellido y nombre. La información recogida en el listado será:</p> <ul style="list-style-type: none"> • Apellidos • Nombre • Teléfono • Edad • Fecha de Nacimiento • Teléfono alternativo • Correo electrónico 					

Req. Software 40: SR.40 - Listado de participantes de actividad

ID: SR.41	Ficha resumen de actividad				
Importancia	Deseable	Estado	Aceptado	Dependencia	RF.L.4
Descripción					
<p>Se podrá generar un informe de una actividad que incluya toda la información recogida en su ficha, junto a los ingresos y gastos, y el listado de participantes, con los atributos especiales definidos.</p>					

Req. Software 41: SR.41 - Ficha resumen de actividad

ID: SR.42	Listados de proyectos por área				
Importancia	Deseable	Estado	Aceptado	Dependencia	RF.L.5
Descripción					
<p>Se podrá generar un listado de proyectos por área de actividad.</p>					

Req. Software 42: SR.42 - Listados de proyectos por área

ID: SR.43	Listados de programas por proyecto				
Importancia	Deseable	Estado	Aceptado	Dependencia	RF.L.6
Descripción					
<p>Se podrá generar un listado de programas por proyecto.</p>					

Req. Software 43: SR.43 - Listados de programas por proyecto

ID: SR.44	Informe de participación				
Importancia	Deseable	Estado	Aceptado	Dependencia	RF.L.7
Descripción					
<p>Se podrá generar un informe de participación por proyecto. En la que recoja el número de participantes en los diferentes programas. Este informe mostrará los índices de participación tanto gráfica como numéricamente.</p>					

Req. Software 44: SR.44 - Informe de participación

ID: SR.45	Listados de participantes por áreas de interés				
Importancia	Deseable	Estado	Aceptado	Dependencia	RF.L.8
Descripción					
Se podrá generar un listado de participantes por áreas de interés.					

Req. Software 45: SR.45 - Listados de participantes por áreas de interés

ID: SR.46	Listado de Voluntarios				
Importancia	Deseable	Estado	Aceptado	Dependencia	RF.L.9
Descripción					
El sistema permitirá generar un listado de los voluntarios de la asociación, ordenados por apellido y nombre.					

Req. Software 46: SR.46 - Listado de Voluntarios

ID: SR.47	Listados en formatos especiales, mailing				
Importancia	Alta	Estado	Aceptado	Dependencia	RF.L.10
Descripción					
El sistema deberá permitir realizar impresiones de etiquetas de mailing, con la información de contacto postal de los participantes, trabajadores y/o colaboradores, a través de varios criterios de búsqueda. El formato de las etiquetas de mailing es el que corresponde al modelo de etiquetas APLI 70x35.					

Req. Software 47: SR.47 - Listados en formatos especiales, mailing

ID: SR.48	Mailing filtrado				
Importancia	Deseable	Estado	Aceptado	Dependencia	RNF.14
Descripción					
Al realizar los listados de mailing, se compararán las direcciones postales, de forma que aquellas que coincidan aparecerán en una única etiqueta, unificando el nombre de los destinatarios.					

Req. Software 48: SR.48 - Mailing filtrado

ID: SR.49	Ficha resumen de participante				
Importancia	Deseable	Estado	Aceptado	Dependencia	RF.L.11
Descripción					
El sistema permitirá imprimir un informe de cada participante, en el que se recoja los datos de contacto, su fotografía, y el historial de participación en la asociación (inscripciones en actividades y programas). Indicando por cada actividad, cuantos recibos tiene pendiente de pago.					

Req. Software 49: SR.49 - Ficha resumen de participante

ID: SR.50	Listado de mailing				
Importancia	Deseable	Estado	Aceptado	Dependencia	RF.L.12
Descripción					
Se podrá generar un listado en excel con las direcciones de participantes, mediante diferentes filtros. Esta característica es necesaria para el envío de publicidad postal a través de terceras empresas.					

Req. Software 50: SR.50 - Listado de mailing

ID: SR.51	Scripts de migración				
Importancia	Alta	Estado	Aceptado	Dependencia	RNF.12
Descripción					
Se generará un proceso de migración de datos de los sistemas existentes al nuevo. Se deberán migrar los datos de contacto de los participantes, su historial de participación, la definición de las áreas y programas, de al menos los últimos 5 años. El proceso deberá evitar la duplicidad de información.					

Req. Software 51: SR.51 - Scripts de migración

ID: SR.52	Multiidioma				
Importancia	Deseable	Estado	Aceptado	Dependencia	RNF.9
Descripción					
El idioma base de las interfaces será el inglés, pero se deberá mostrar alternativamente en idioma castellano. Cada usuario podrá seleccionar que idioma prefiere utilizar.					

Req. Software 52: SR.52 - Multiidioma

ID: SR.53	Compatibilidad hacia atrás				
Importancia	Deseable	Estado	Aceptado	Dependencia	RNF.8
Descripción					
El sistema se construye sobre Openbravo ERP, hay que conservar los modelos de datos para que sean compatibles con toda la funcionalidad existente, para poder ser integrada en el momento que sea oportuno.					

Req. Software 53: SR.53 - Compatibilidad hacia atrás

ID: SR.54	Arquitectura web				
Importancia	Alta	Estado	Aceptado	Dependencia	RNF.3, RNF.2, RNF.11
Descripción					
La arquitectura del sistema debe ser tipo web, para permitir su acceso desde diferentes sistemas operativos, y desde diferentes ubicaciones geográficas.					
Comentarios					
Característica del sistema Openbravo.					

Req. Software 54: SR.54 - Arquitectura web

ID: SR.55	Sistema multisesión				
Importancia	Alta	Estado	Aceptado	Dependencia	RNF.1
Descripción					
Deberá ser posible que varios usuarios trabajen a la vez con el sistema sin conflictos y garantizando la disponibilidad y consistencia de la información para todos los usuarios.					

Req. Software 55: SR.55 - Sistema multisesión

ID: SR.56	Sistema de ayuda				
Importancia	Deseable	Estado	Aceptado	Dependencia	RNF.6
Descripción					
El sistema ofrecerá información de ayuda accesible desde todas las interfaces, para guiar al usuario acerca de cómo utilizar las diferentes funcionalidades.					
Comentarios					
Característica del sistema Openbravo.					

Req. Software 56: SR.56 - Sistema de ayuda

ID: SR.57	Curso online				
Importancia	Deseable	Estado	Aceptado	Dependencia	RNF.5
Descripción					
Como parte del proyecto se elaborará un curso formativo online que guíe a los nuevos usuarios en el proceso de aprendizaje del nuevo sistema. Este recogerá la funcionalidades más habituales con las que un usuario normal puede encontrarse cada día en el trabajo con la aplicación.					

Req. Software 57: SR.57 - Curso online

ID: SR.58	Interfaz amigable				
Importancia	Deseable	Estado	Aceptado	Dependencia	RNF.5
Descripción					
Las interfaces de la aplicación usarán la tecnología disponible para mejorar la experiencia del usuario y mejorar su productividad. Estará construida con hojas de estilo y AJAX.					
Comentarios					
Característica del sistema Openbravo.					

Req. Software 58: SR.58 - Interfaz amigable

ID: SR.59	Backup automatizado				
Importancia	Alta	Estado	Aceptado	Dependencia	RNF.10, RNF.4
Descripción					
Para garantizar la no pérdida de información se realizará cada día un backup automatizado. Este backup se realizará con las herramientas de planificación del sistema operativo. Para evitar interferir con el trabajo de los usuarios, este proceso deberá realizarse por la noche.					

Req. Software 59: SR.59 - Backup automatizado

Diseño detallado

A continuación se analizan los principales elementos que han influido en el proyecto, comenzando por el diseño y desarrollo del sistema, y continuando con su implantación, mantenimiento y la formación ofrecida a los usuarios destinatarios.

1 Sistema

En esta sección se repasan los criterios seguidos en el diseño del sistema, analizando uno a uno los módulos añadidos y modificados del sistema base.

1.1 *Convenio de diseño*

Puesto que el sistema YMCA Openbravo ha sido desarrollado sobre la plataforma Openbravo, abierta a múltiples desarrollos y soportado por varias empresas, se han seguido sus recomendaciones para asegurar su mantenibilidad futura y evitar que nuevos desarrollos produzcan conflictos con el desarrollo de este proyecto.

Como establecen, cada clase implementada y cada tabla creada o registro insertado en tablas existentes, deben contener un identificador del desarrollo. Esta metodología es común también en otros grandes sistemas corporativos como SAP o Siebel. En el caso de Openbravo, cada desarrollo de terceros debe incluir las letras CUS (custom) y las tres primeras letras del cliente, en este proyecto se ha utilizado YMC (comienzo de las siglas YMCA).

Así como se puede observar en los diagramas de clases y modelos de datos cada clase, tabla o registro modificado incluye el prefijo “cus_ymc”. Este identificador resulta útil para distinguir las clases modificadas en el proyecto, y facilitar en el futuro las posibles migraciones a nuevas versiones.

1.2 Módulos nuevos

En esta sección se analizan los diferentes módulos que han sido añadidos al sistema, que no cubría la aplicación original y eran requeridos por los requisitos de usuario.

1.2.1 Gestión de Áreas

El objetivo de este módulo es el de definir conjuntos que definan las diferentes áreas de actividad de la asociación.

Bajo este conjunto se agruparán los proyectos, y permitirá a la larga obtener estadísticas de productividad de las diferentes áreas de la organización.

Su diseño es bien sencillo y se basa en el modelo de datos que se muestra en la imagen Ilustración 8, a partir de él se construye la ventana con el diccionario de la aplicación.

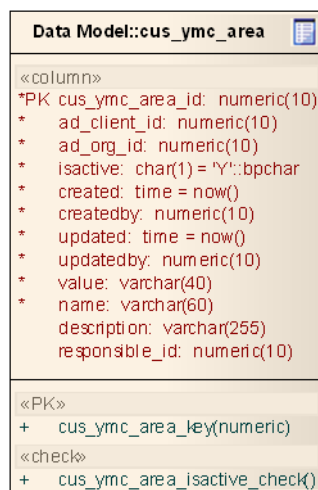
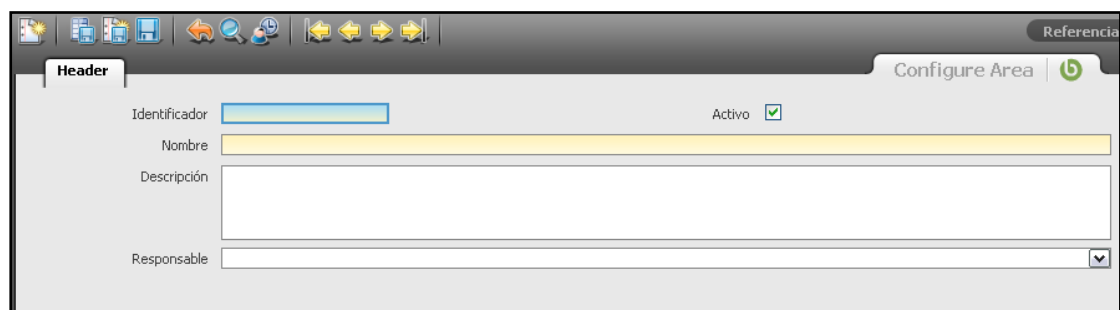


Ilustración 8: Modelo de datos del Área

La interfaz disponible se muestra a continuación:



The screenshot shows a web-based interface for configuring an area. It has a header bar with a 'Referencia' button and a 'Configure Area' tab. The main form contains the following fields:

- Identificador:** A text input field.
- Nombre:** A text input field.
- Descripción:** A large text area.
- Responsable:** A dropdown menu.
- Activo:** A checkbox that is currently checked.

Ilustración 9: Interfaz de inserción de Áreas

1.2.2 Gestión de Proyectos

La gestión de proyectos permite definir otro nivel de agrupamiento que será utilizado para agrupar los programas que se realizan en cada una de las sedes de la organización, para realizar un proyecto común. Por ejemplo “Campamentos de Verano 2009” sería un proyecto donde cada sede crearía su campamento por medio de un programa.

Con vista a en un futuro poder realizar un balance de los gastos e ingresos globales de un proyecto, se definieron además valores económicos.

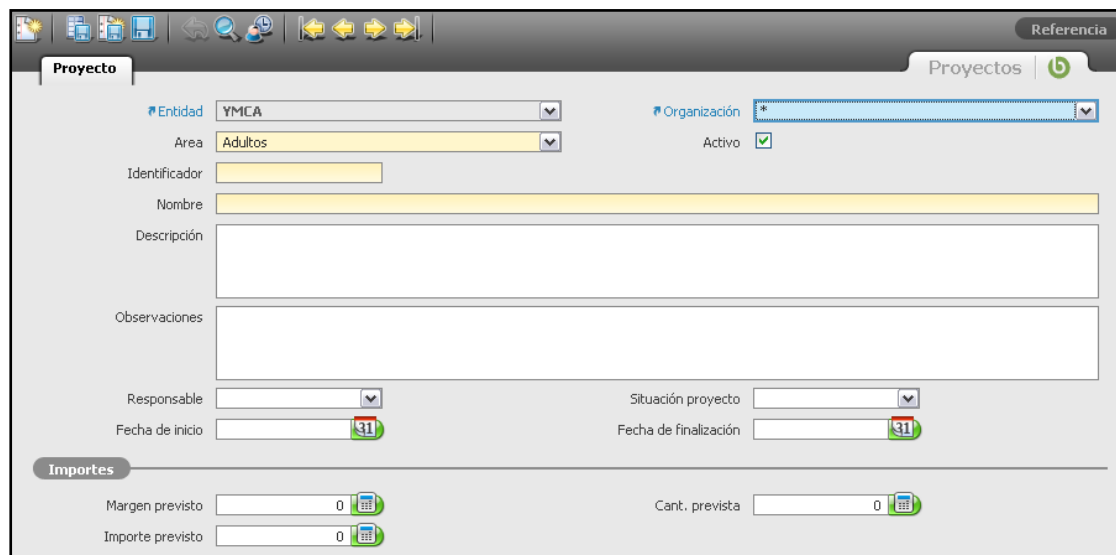


Ilustración 10: Interfaz de inserción de Proyectos

El modelo de datos que define el proyecto se muestra en la Ilustración 11.

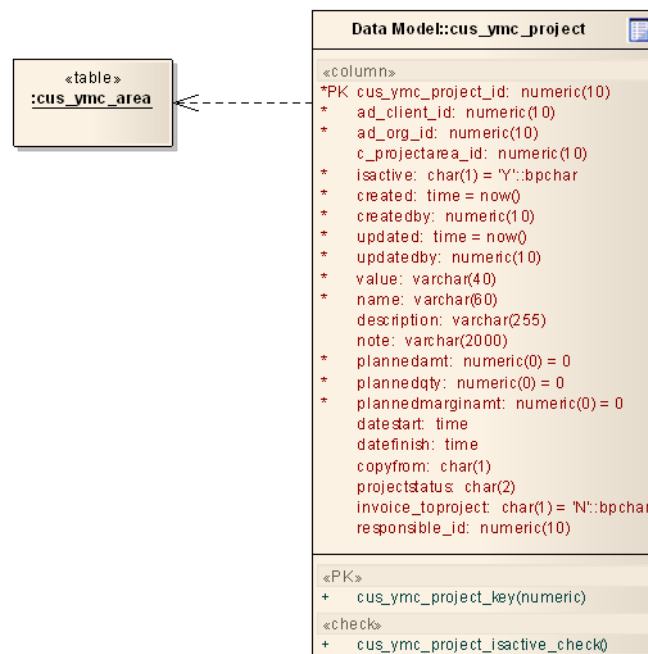


Ilustración 11: Modelo de datos de proyecto

1.2.3 Gestión de Programas

Los programas son la realización concreta de un proyecto en cada una de las sedes, o en varias a la vez.

La definición de un programa incluye las fechas de realización, que define el periodo en que un participante podrá ser inscrito en dicho programa y durante el tiempo que se facturará.

Cada programa puede tener un coste de participación y este coste puede ser puntual o mensual, en caso de ser mensual, cada mes se podrán generar recibos nuevos por cada inscripción de los participantes.

Uno de los rasgos particulares del programa es la posibilidad de incluir otros parámetros personalizados, estos parámetros serán utilizados a la hora de realizar la inscripción de cada participante, por ejemplo puede incluirse la parada donde un autocar de ruta recogerá a los participantes en un campamento urbano de verano.

La característica de atributos personalizados se describe más a fondo en el apartado 1.3.4 de esta sección.

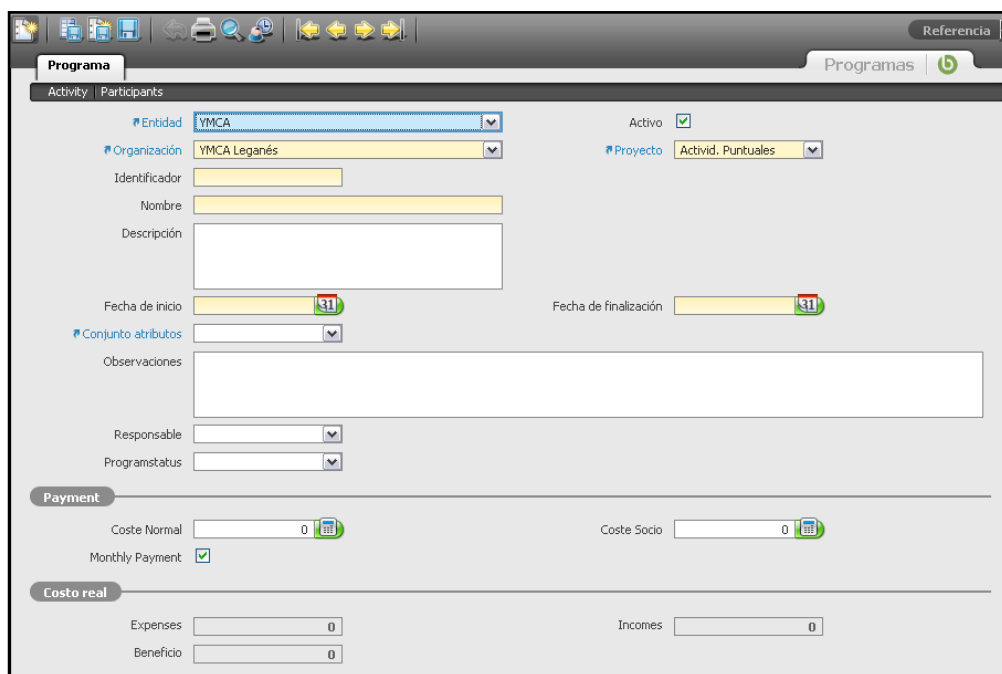


Ilustración 12: Interfaz de inserción de Programas

El modelo de datos sobre el que se implementa la definición de los programas se muestra en la Ilustración 13.

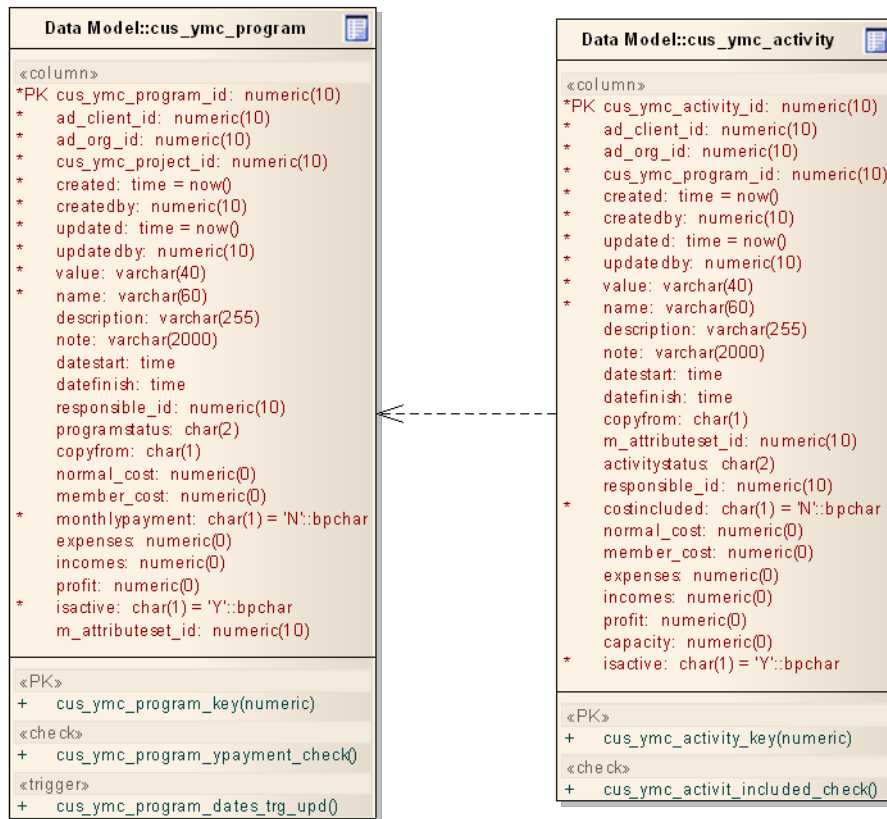


Ilustración 13: Modelo de datos de programas y actividades

1.2.4 Gestión de Actividades

Las actividades es el último nivel de agrupación, y permite opcionalmente, dividir cada programa en diferentes actividades. Por ejemplo en un programa de ocio y tiempo libre, podría crearse una actividad puntual para la realización de una excursión. Esta actividad puede tener un coste extra, siendo posible realizar un nuevo recibo de cobro, o estar incluido su coste en el del programa, lo que supondrá un coste 0.

Tanto en Programas como Actividades, se proporcionan dos precios uno para los participantes normales, y otro para los socios, esta característica requerida, ya existía en la aplicación anterior.

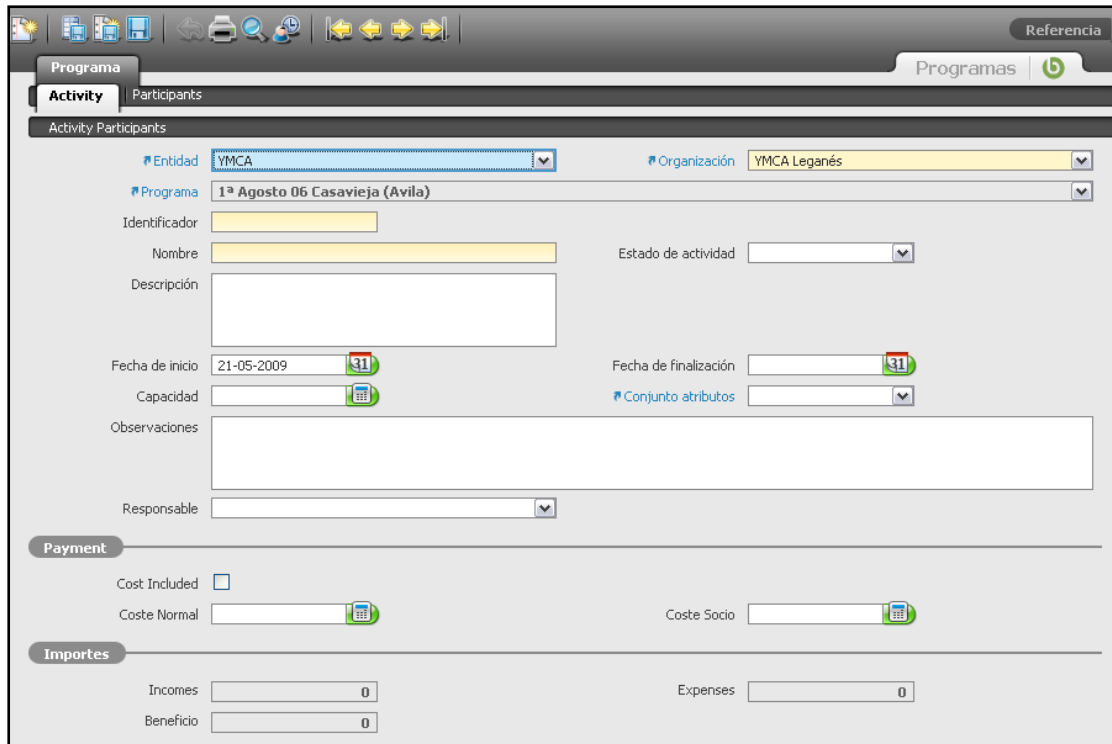


Ilustración 14: Interfaz de inserción de Actividades

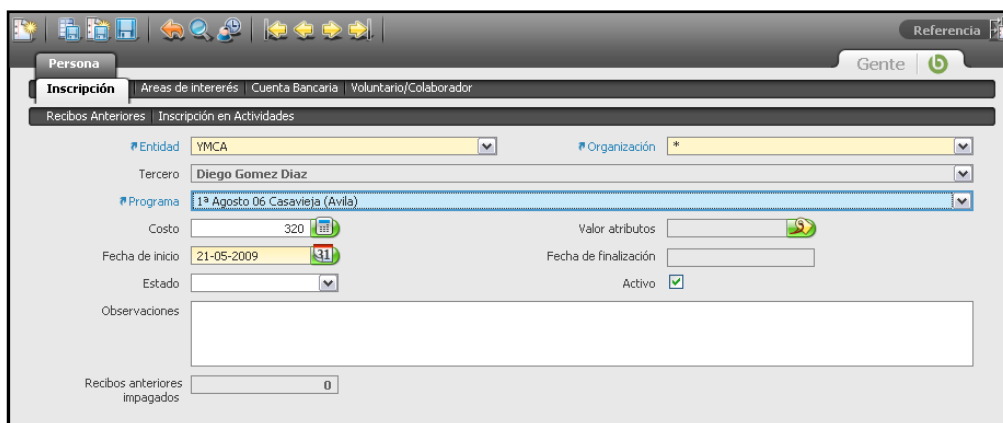
En la Ilustración 13 se muestra el modelo con que se implementa la definición de las Actividades, junto con el Programa al que pertenecería.

1.2.5 Gestión de Inscripciones en Programas y Actividades

Una vez configurado un programa y dado de alta un participante, es posible realizar su inscripción en dicho programa, por medio del módulo de gestión de inscripciones. Esta característica es común para los Programas y las Actividades, pero su estructura es algo diferente.

En ambas posibilidades se incluye la característica de completar además de los campos habituales, los personalizados que han sido definidos para el programa o actividad.

En caso de que el programa o actividad supongan algún coste de participación, tras definir la inscripción se permitirá crear el recibo de cobro, característica descrita en el siguiente apartado.



Persona | **Inscripción** | Áreas de interés | Cuenta Bancaria | Voluntario/Colaborador

Recibos Anteriores | Inscripción en Actividades

Entidad: YMCA | Organización: *

Tercero: Diego Gomez Diaz

Programa: 1º Agosto 06 Casavieja (Avila)

Costo: 320 | Valor atributos: [icon]

Fecha de inicio: 21-05-2009 | Fecha de finalización: []

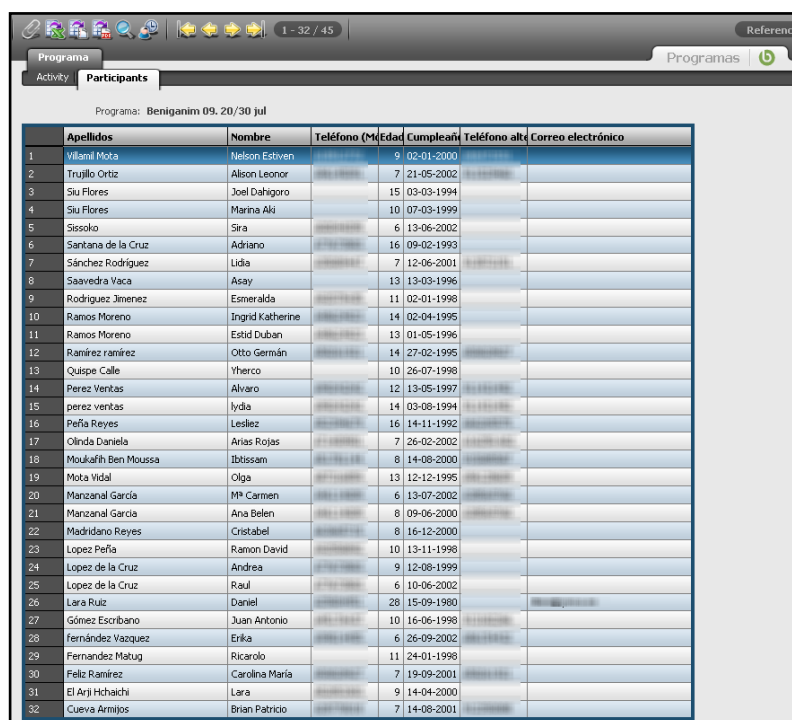
Estado: [] | Activo: ☒

Observaciones: []

Recibos anteriores impagados: 0

Ilustración 15: Interfaz de Inscripción de Participantes

Al inscribir a los participantes es posible obtener un listado de participación directamente desde las fichas del programa o actividad.



Programa | **Activity** | **Participants**

Programa: Beniganim 09. 20/30 jul

	Apellidos	Nombre	Teléfono (Móvil)	Edad	Cumpleaños	Teléfono alt	Correo electrónico
1	Villanil Mota	Nelson Estiven	[]	9	02-01-2000	[]	[]
2	Trujillo Ortiz	Alison Leonor	[]	7	21-05-2002	[]	[]
3	Siu Flores	Joel Dahigoro	[]	15	03-03-1994	[]	[]
4	Siu Flores	Marina Ali	[]	10	07-03-1999	[]	[]
5	Sissoko	Sira	[]	6	13-06-2002	[]	[]
6	Santana de la Cruz	Adriano	[]	16	09-02-1993	[]	[]
7	Sánchez Rodríguez	Lidia	[]	7	12-06-2001	[]	[]
8	Saavedra Vaca	Asay	[]	13	13-03-1996	[]	[]
9	Rodriguez Jimenez	Esmeralda	[]	11	02-01-1998	[]	[]
10	Ramos Moreno	Ingri Katherine	[]	14	02-04-1995	[]	[]
11	Ramos Moreno	Estid Duban	[]	13	01-05-1996	[]	[]
12	Ramirez ramirez	Otto Germán	[]	14	27-02-1995	[]	[]
13	Quispe Calle	Yherco	[]	10	26-07-1998	[]	[]
14	Perez Ventas	Alvaro	[]	12	13-05-1997	[]	[]
15	perez ventas	lydia	[]	14	03-08-1994	[]	[]
16	Peña Reyes	Lesliez	[]	16	14-11-1992	[]	[]
17	Olinda Daniela	Arias Rojas	[]	7	26-02-2002	[]	[]
18	Moukafih Ben Moussa	Ibtissam	[]	8	14-08-2000	[]	[]
19	Mota Vidal	Olga	[]	13	12-12-1995	[]	[]
20	Manzanal García	Mª Carmen	[]	6	13-07-2002	[]	[]
21	Manzanal García	Ana Belen	[]	8	09-06-2000	[]	[]
22	Madridano Reyes	Cristabel	[]	8	16-12-2000	[]	[]
23	Lopez Peña	Ramon David	[]	10	13-11-1998	[]	[]
24	Lopez de la Cruz	Andrea	[]	9	12-08-1999	[]	[]
25	Lopez de la Cruz	Raul	[]	6	10-06-2002	[]	[]
26	Lara Ruiz	Daniel	[]	28	15-09-1980	[]	[]
27	Gómez Escribano	Juan Antonio	[]	10	16-06-1998	[]	[]
28	fernández Vazquez	Enika	[]	6	26-09-2002	[]	[]
29	Fernandez Matug	Ricardo	[]	11	24-01-1998	[]	[]
30	Feliz Ramirez	Carolina Maria	[]	7	19-09-2001	[]	[]
31	El Arji Hchaichi	Lara	[]	9	14-04-2000	[]	[]
32	Cueva Armitos	Brian Patricio	[]	7	14-08-2001	[]	[]

Ilustración 16: Listado de Inscripciones

Además hay definido un proceso en Background (Ilustración 17) que comprueba al finalizar cada mes cuantas inscripciones están pendientes de cobro. Este proceso almacena en cada inscripción un recibo pendiente de cobro y habilita los botones en las inscripciones para generar los recibos del mes actual.

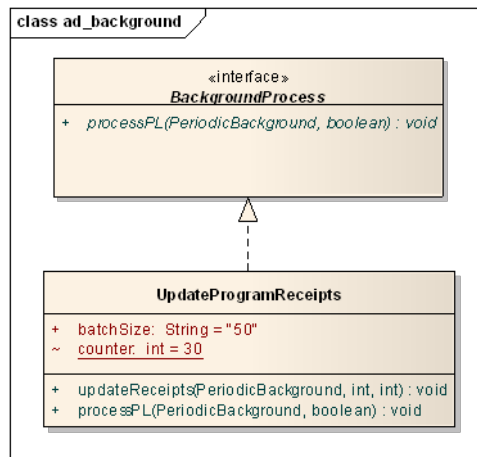


Ilustración 17: Clases que implementa el proceso de background

En la Ilustración 19 aparece el modelo de datos utilizado para la realización de las inscripciones. Cada inscripción está asociada con su programa o actividad. En el caso de los programas se utiliza además un registro extra que es utilizado para contabilizar los recibos generados para ese programa, de modo que es posible llevar control de los recibos expedidos durante todos los meses de duración de proyecto.

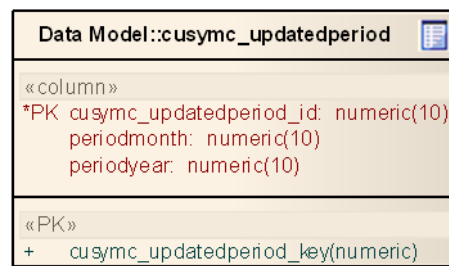


Ilustración 18: Control del proceso en background

Se lleva un control en la base de datos para evitar que este proceso se ejecute más de una vez al mes. El modelo utilizado aparece en la Ilustración 18.

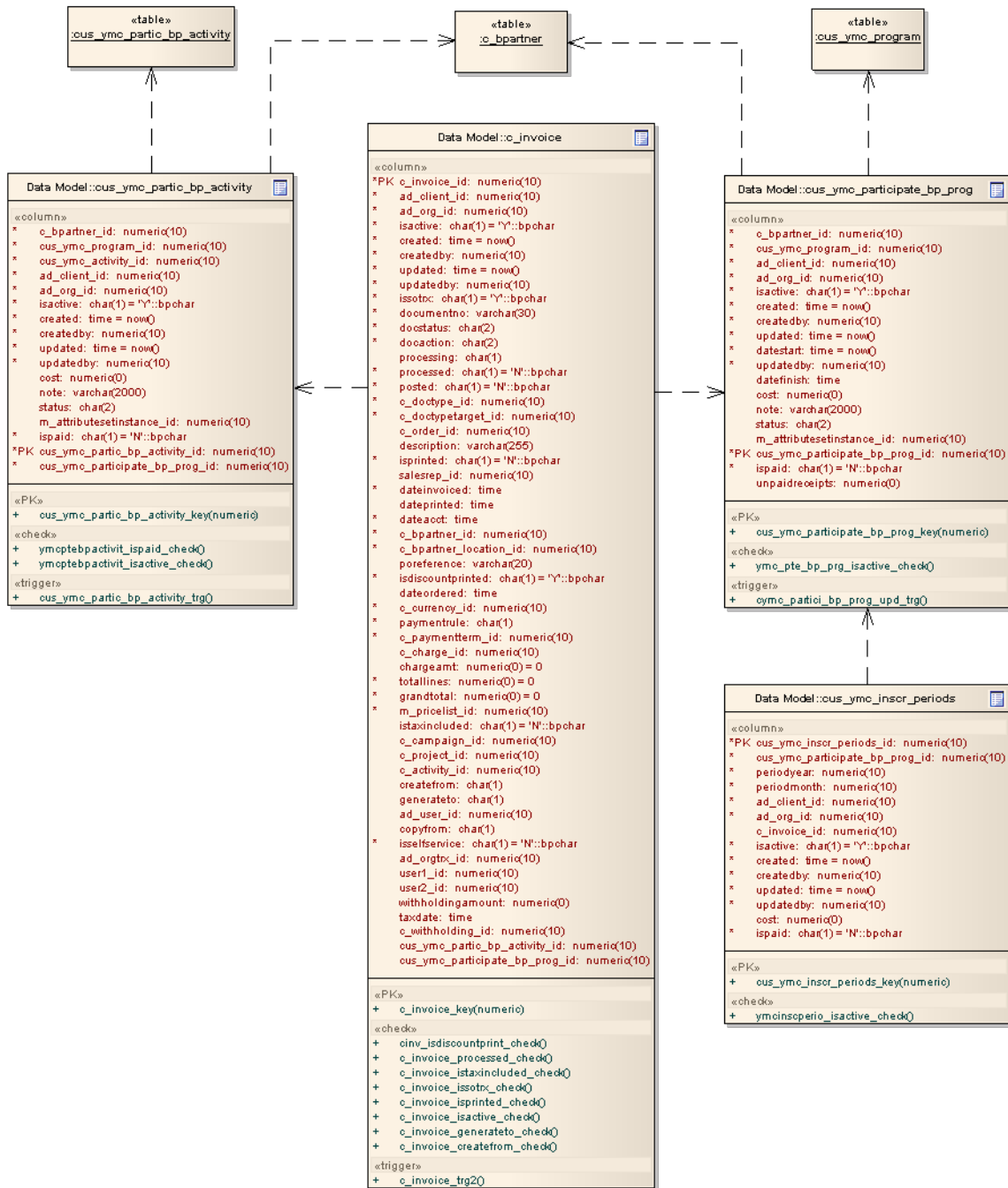


Ilustración 19: Modelo de datos de las inscripciones

Existe además otro elemento en este componente a tener en cuenta, es el que se muestra en la Ilustración 20. Estas clases permiten actualizar el precio en la interfaz de la inscripción al seleccionar uno u otro programa o actividad.

Los callout en Openbravo se integran con la vista de forma que un cambio en la lista desplegable de programas y/o actividades, actualiza otros campos de la interfaz.

class ad_callouts	
CUS_YMC_UpdateProgramCost	<i>HttpSecureAppServlet</i>
- serialVersionUID: long = 1L {readOnly}	
~ ZERO: BigDecimal = new BigDecimal(0.0) {readOnly}	
+ init(ServletConfig) : void	
+ doPost(HttpServletRequest, HttpServletResponse) : void	
~ printPage(HttpServletResponse, VariablesSecureApp, String, String, String, String, String, String) : void	
CUS_YMC_UpdateActivityCost	<i>HttpSecureAppServlet</i>
- serialVersionUID: long = 1L {readOnly}	
~ ZERO: BigDecimal = new BigDecimal(0.0) {readOnly}	
+ init(ServletConfig) : void	
+ doPost(HttpServletRequest, HttpServletResponse) : void	
~ printPage(HttpServletResponse, VariablesSecureApp, String, String, String, String, String, String) : void	

Ilustración 20: Callout para la actualización de precios

1.2.6 Generación de Recibos de Cobro

La característica por excelencia que tenía la aplicación anterior y que motivaba el uso del sistema pese a las continuas caídas y fallos, era la posibilidad de generar recibos de cobro por la participación en actividades.

En una de las reuniones mantenidas con la gerencia esta característica se iba a dejar fuera por no mantener finalmente el módulo financiero. Sin embargo analizando la importancia que los trabajadores le daban a esta característica, y pensando en que quizá su falta haría poco práctico el sistema para los trabajadores en su día a día finalmente se optó por mantenerla.

Al realizar las inscripciones de programas o actividades, se muestra un botón que permite la realización automática de un recibo con un formato idéntico al de la aplicación antigua que incluye los campos que indican la actividad o programa, el nombre de participante, su dirección y la cantidad del ingreso, que coincide con el coste de participación del programa.

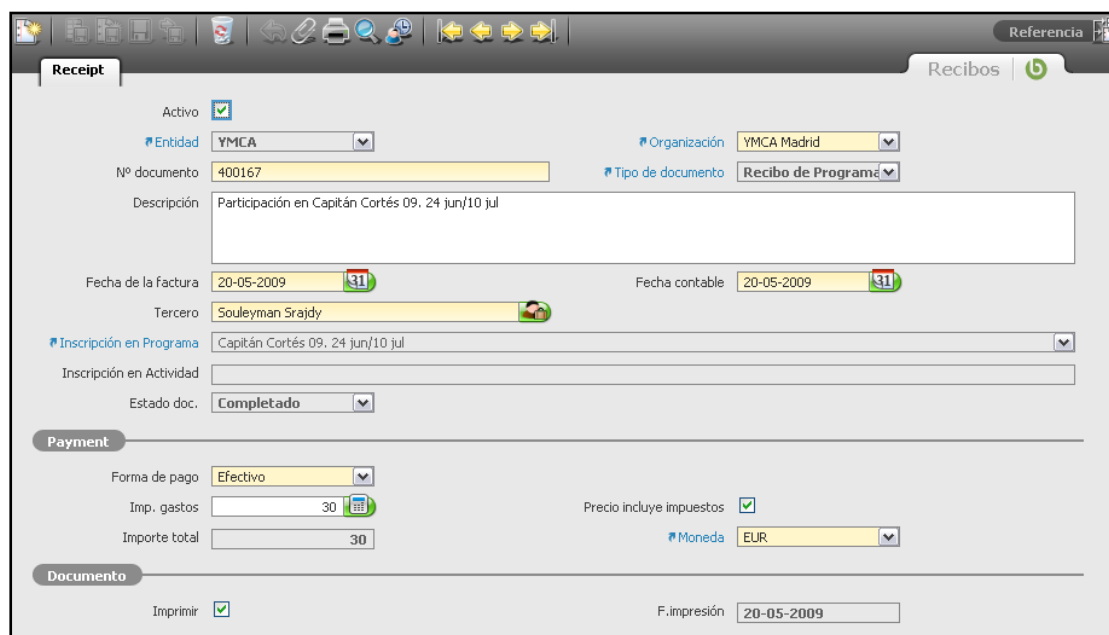



Ilustración 21: Interfaz de edición de Recibos generados

Para evitar errores, una vez generado un recibo de una inscripción, el botón que permite generar uno nuevo desaparece, hasta el mes siguiente, si la actividad es de pago mensual. En el caso de los programas de pago mensual, si algún recibo no ha sido generado y transcurre todo un mes desde la inscripción, ese recibo queda acumulado para poder generarse posteriormente. Siendo accesible mediante la pestaña de Recibos Anteriores en la ventana de inscripciones.

El recibo es generado en un archivo PDF, para que a continuación sea impreso y entregado al participante.

Nº Recibo:	400066		YMCA Madrid c/ Alba de Tormes, 10 28025 Madrid MADRID-Spain - Fax: 914610764 Telf: 915251676 e-mail: madrid@ymca.es www.ymca.es
Asociación declarada de Utilidad Pública el 10 de Abril de 1981. Afiliada a la Alianza Mundial de YMCA's Fundada en Londres en 1988. C.I.F.: G-28659308			
Hemos recibido de	DiegoGomez Diaz		
con domicilio en	C/ Rio Ebro, 16 3ºa28935Mostoles		
la cantidad de			euros.
en concepto del resarcimiento del gasto por	Participación en Curso iniciación a YMCA Openbravo		
Son	0,00	Euros.	miércoles, 22 abril 2009

Copia del recibo para el participante.

Ilustración 22: Formato del recibo generado

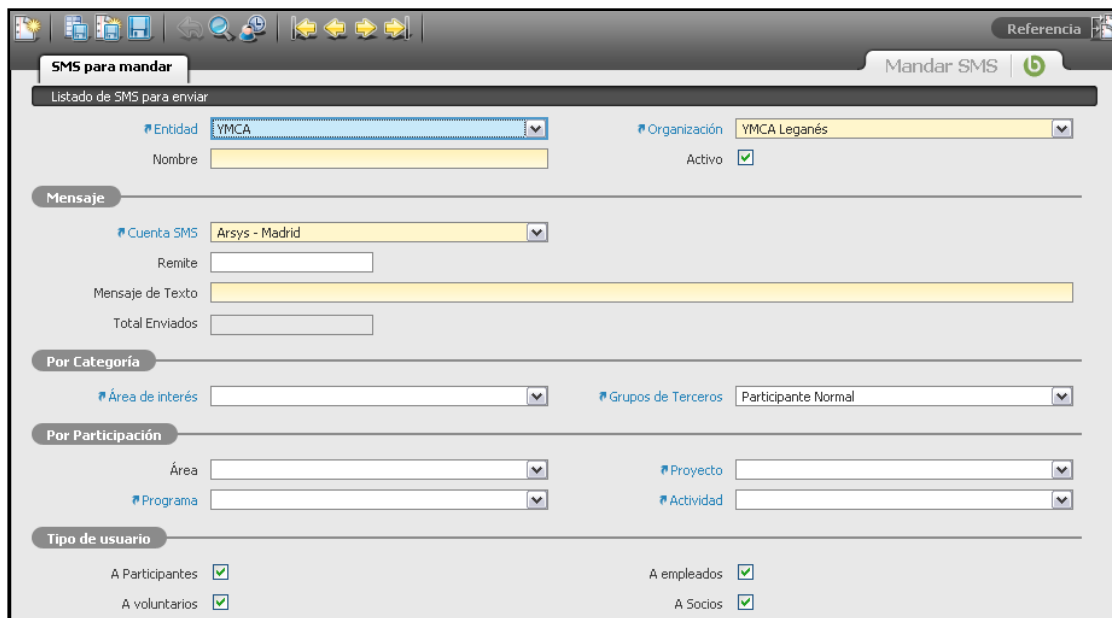
En la Ilustración 19 aparece la tabla 'c_invoice' de la base de datos utilizada para la generación de recibos. El motivo de utilizar esta tabla existente de Openbravo ERP es para facilitar la incorporación futura de este elemento a la contabilidad. En Openbravo ERP, toda factura, recibo, albarán u orden de compra se almacena en esta tabla y del valor de c_doctype_id depende el tipo de documento utilizado.

Además la utilización de 'c_invoice' permite utilizar una característica de Openbravo que controla la numeración de los documentos, así es posible configurar la numeración asignada a los recibos siendo posible cambiar su formato y garantizando su unicidad.

1.2.7 Notificación por SMS

Pensando en las características que podrían dar valor a la aplicación se decidió implementar un módulo capaz de comunicarse con un servicio de envío de SMS externo para gestionar el envío de mensajes a móviles desde la aplicación.

Así se puede aprovechar la información existente en la aplicación y realizar envíos masivos dirigidos a ciertos sectores, mediante la aplicación de filtros y la inclusión manual. Así por ejemplo se podría mandar un SMS a todos los participantes de un programa para notificarle un cambio en la planificación, o a los interesados en cierta área para anunciarles un nuevo programa de su interés.



The screenshot shows a web application interface titled "SMS para mandar". It includes a toolbar at the top with icons for file operations and a "Referencia" button. The main content area is divided into several sections:

- Listado de SMS para enviar:** Contains dropdowns for "Entidad" (set to YMCA) and "Organización" (set to YMCA Leganés), a "Nombre" input field, and an "Activo" checkbox (checked).
- Mensaje:** Contains a "Cuenta SMS" dropdown (set to Arsys - Madrid), a "Remite" input field, a "Mensaje de Texto" input field, and a "Total Enviados" input field.
- Por Categoría:** Contains a dropdown for "Área de interés" and a dropdown for "Grupos de Terceros" (set to Participante Normal).
- Por Participación:** Contains dropdowns for "Área", "Programa", "Proyecto", and "Actividad".
- Tipo de usuario:** Contains checkboxes for "A Participantes", "A voluntarios", "A empleados", and "A Socios", all of which are checked.

Ilustración 23: Interfaz de envío de Mensajes de móviles

El módulo ha sido diseñado para permitir usar diferentes servicios de mensajería, aunque inicialmente sólo está disponible uno capaz de comunicarse con el servicio de envío de SMS de Arsys (servicio que tenía contratado la organización en el momento del desarrollo del sistema). En la Ilustración 24 se puede ver la relación existente entre los diferentes elementos de este módulo.

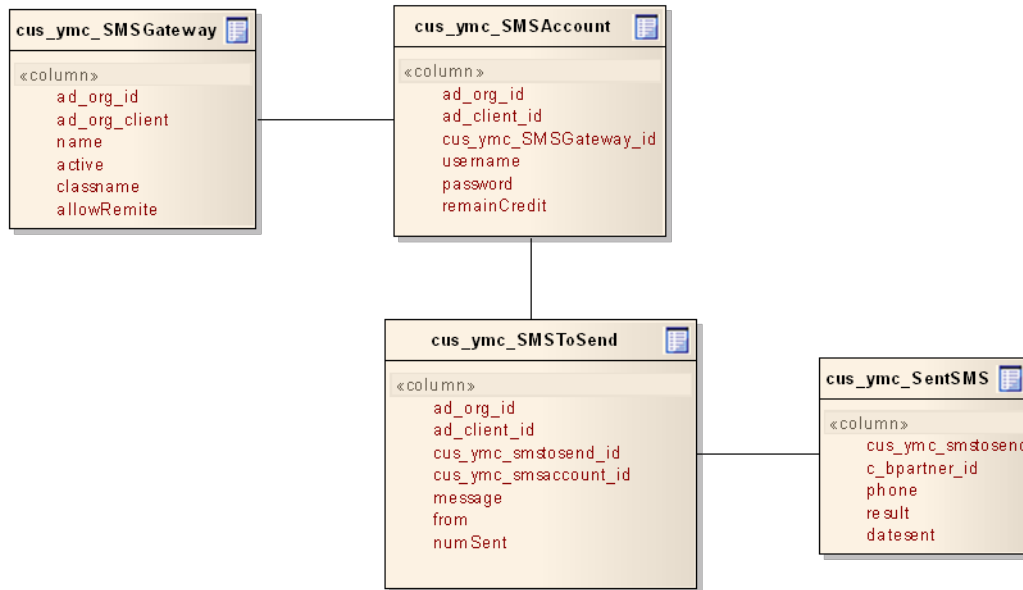


Ilustración 24: Modelo módulo de envío de SMS

Para un servicio de mensajería se pueden tener además, configuradas diferentes cuentas, cada una generalmente con un paquete disponible de SMS. Puesto que el uso de estos sistemas suelen tener un coste asociado, se mantiene un registro de todos los mensajes enviados, así como de los mensajes recibidos por el operador, indicando el estado de envío de cada mensaje.

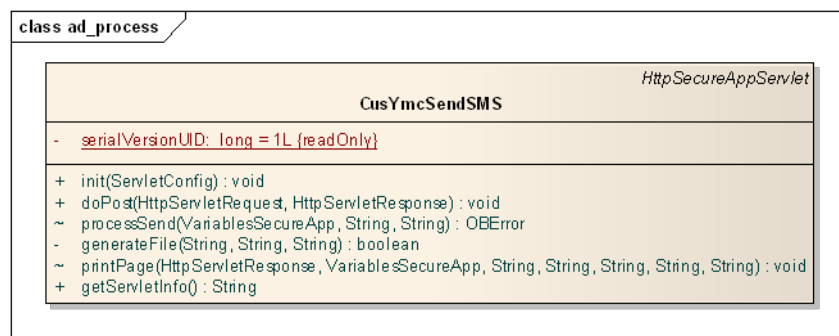


Ilustración 25: Clase de envío de SMS

En la Ilustración 25 se muestra la clase que implementa el proceso encargado de atender la petición de envío de SMS, aunque internamente utiliza las clases de envío definidas en la Ilustración 26, que son capaces de comunicarse con el servidor de envío.

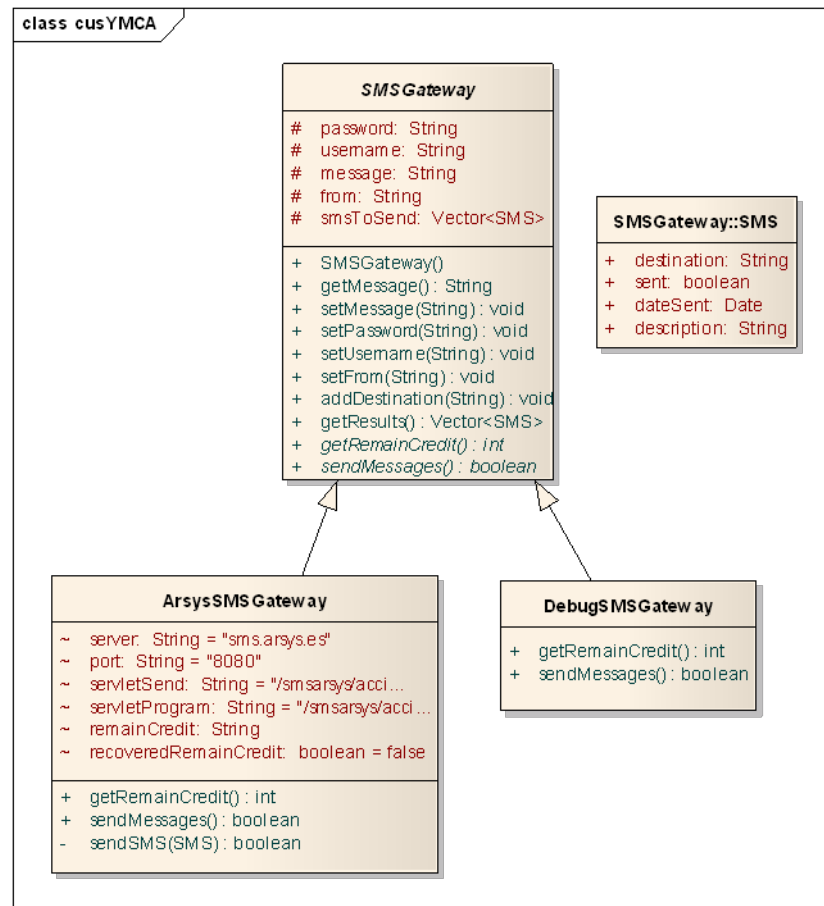


Ilustración 26: Clases gateway de envío de SMS

1.2.8 Informes de Mailing

Una de las características que ofrecía la aplicación antigua era la posibilidad de crear etiquetas con las direcciones de los participantes para la elaboración de Mailings. Para mantener esa funcionalidad se creó este informe especial que permite realizar filtrado por diferentes criterios.

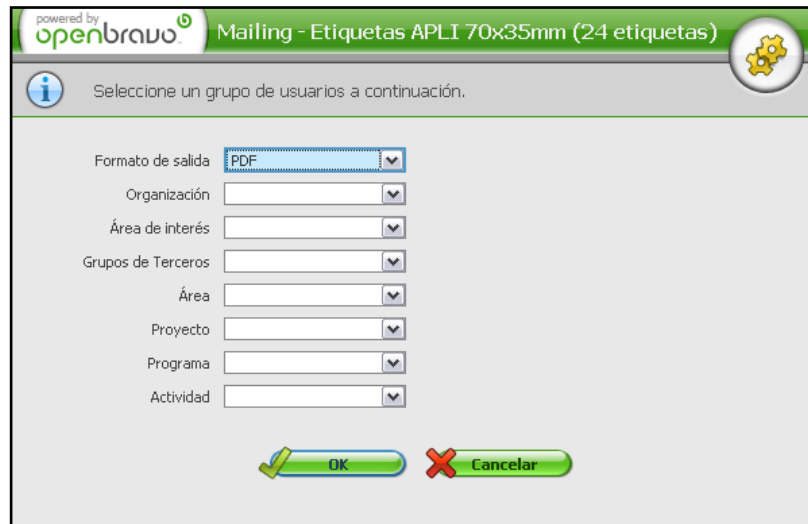


Ilustración 27: Interfaz de filtro de creación de etiquetas de Mailing

1.3 Funcionalidad cambiada o expandida del ERP

En esta sección se cubren los componentes que han sido construidos a base de la reutilización y expansión de parte de sistema existente.

1.3.1 Compatibilidad con sistema Openbravo ERP

En el momento del análisis y diseño del sistema se tuvo en mente la necesidad de mantener la compatibilidad total con el sistema ERP construido sobre Openbravo, con el objetivo de en un futuro poder comenzar a utilizar las herramientas de este ERP sin la necesidad de realizar ningún cambio en el sistema. Además se buscaba que todo componente reutilizado mantuviese su funcionamiento original.

Para cumplir este objetivo, en lugar de modificar los componentes existentes se añadieron otros heredando las características de los primeros, manteniendo cada módulo intacto.

Por otro lado el modelo de datos fue ampliado manteniendo campos aunque no se utilizasen, y aunque requiriese en realidad un cambio de concepto.

Un claro ejemplo de este hecho fue el diseño del modelo de datos de participante a partir del existente Business Partner. En Openbravo ERP, un colaborador o tercero es una empresa con una serie de personas de contacto. En YMCA Openbravo, cada tercero es en realidad una sola persona asociada a un solo usuario, que incluye los datos personales de contacto.

Pero mientras que en Openbravo ERP, los datos de contacto no eran siempre importantes, en YMCA Openbravo sí que lo era, por eso se definió que al crear un tercero se crease un usuario asociado, y sólo uno. Siempre que se acceda desde la interfaz del módulo de Participantes, que accede en realidad a tres tablas por medio de una vista, y modifica los datos a través de modificadores en la vista.

1.3.2 Gestión de Participantes

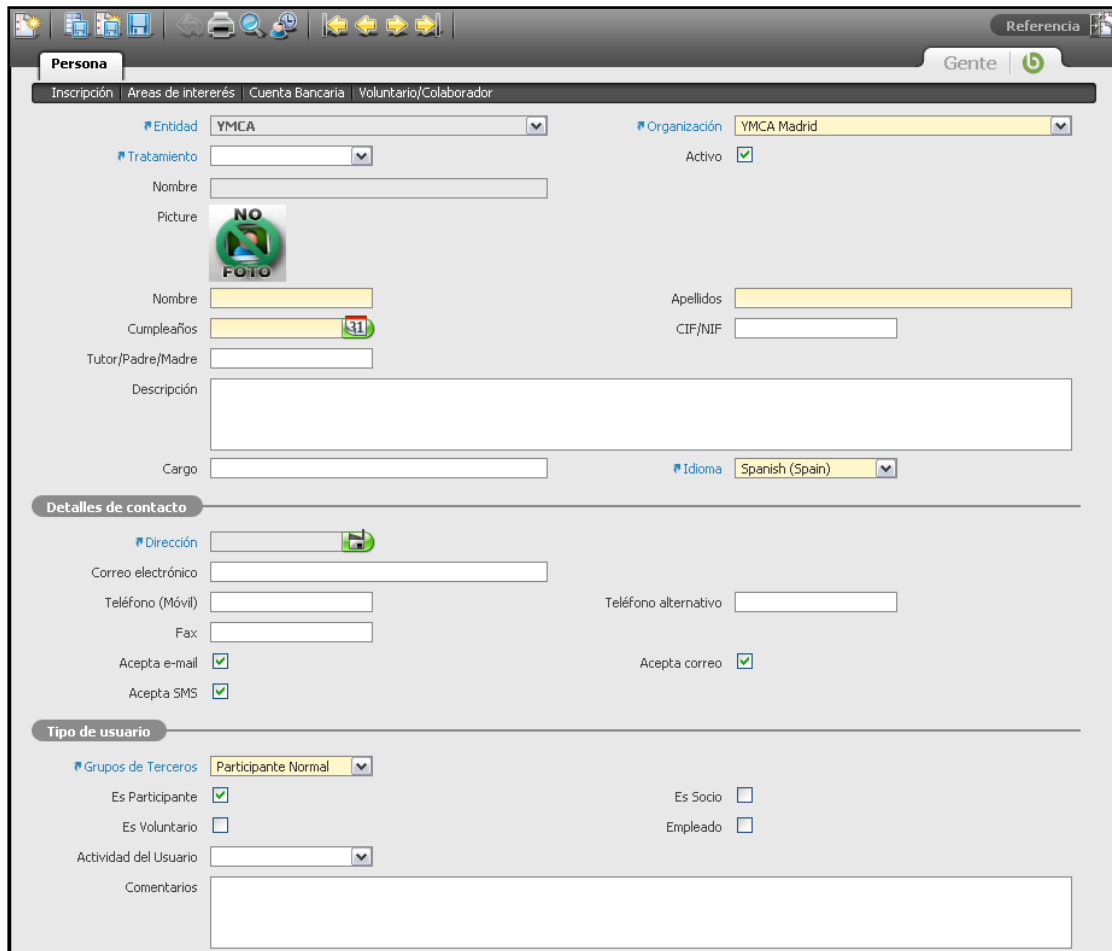
Como se ha introducido en el apartado anterior, la gestión de participantes nace mediante la reutilización y expansión de modelos de datos existentes en Openbravo ERP (Business Partner, User y Location), en una sola ventana a través de la creación de una vista.

Este elemento se creó así para mantener la compatibilidad con el resto de los módulos, y bajo recomendación de técnicos de la empresa Openbravo.

Como es el proceso más crucial en la aplicación, el diseño e implementación del modelo de datos fue una de las tareas que más tiempo consumió en el proyecto, ya que fueron necesarias varias optimizaciones y creaciones de índices específicos en la base de datos.

Sin embargo, si analizamos otros proyectos, como el reciente Gesmon (11), este modelo podría haberse creado de cero, optimizando así el uso de la base de datos, y seguramente su acceso.

A parte se crearon otros datos, como los datos formativos de un colaborador.



The screenshot displays the 'Persona' (Person) form in a web application. The form is organized into several sections:

- Header:** Includes a 'Referencia' button and a 'Gente' tab.
- Navigation:** Tabs for 'Inscripción', 'Áreas de interés', 'Cuenta Bancaria', and 'Voluntario/Colaborador'.
- Personal Data:**
 - Entidad:** Dropdown menu set to 'YMCA'.
 - Organización:** Dropdown menu set to 'YMCA Madrid'.
 - Tratamiento:** Dropdown menu.
 - Nombre:** Text input field.
 - Picture:** Placeholder image with a 'NO FOTO' watermark.
 - Nombre:** Text input field.
 - Cumpleaños:** Date picker set to '31'.
 - Tutor/Padre/Madre:** Text input field.
 - Descripción:** Large text area.
 - Cargo:** Text input field.
 - Idioma:** Dropdown menu set to 'Spanish (Spain)'.
- Detalles de contacto:**
 - Dirección:** Text input field with a location icon.
 - Correo electrónico:** Text input field.
 - Teléfono (Móvil):** Text input field.
 - Teléfono alternativo:** Text input field.
 - Fax:** Text input field.
 - Acepta e-mail:** Checked checkbox.
 - Acepta correo:** Checked checkbox.
 - Acepta SMS:** Checked checkbox.
- Tipo de usuario:**
 - Grupos de Terceros:** Dropdown menu set to 'Participante Normal'.
 - Es Participante:** Checked checkbox.
 - Es Voluntario:** Unchecked checkbox.
 - Es Socio:** Unchecked checkbox.
 - Empleado:** Unchecked checkbox.
 - Actividad del Usuario:** Dropdown menu.
 - Comentarios:** Large text area.

Ilustración 28: Interfaz de edición de Participantes

1.3.3 Nueva vista de Fotografías de Usuarios

La plataforma Openbravo permite crear ventanas por medio de dos modos, una es programándola entera definiendo las páginas jsp, y los controladores en Java, o por medio del diccionario de datos de la aplicación, lo que permite generar ventanas automáticas a partir de la definición del modelo de datos, y del tipo de vista definida para cada campo (fecha, texto, campo booleano, lista, etc.).

Puesto que el módulo de gestión de participantes era muy complejo y contenía mucha información asociada, se trató de usar el método de construcción automática. Pero había características no cubiertas por la aplicación, y una era la de permitir subir fotografías o archivos como tipo de dato, ya que no existe ningún componente capaz de hacer esto, el más parecido es uno que es capaz de mostrar una fotografía a partir de una URL que apunta a la imagen.

Puesto que este componente se quedaba corto para el uso que debían dar los trabajadores de la asociación al dar de alta a los participantes se creó un nuevo componente visual capaz de gestionar esta característica.



Ilustración 29: Inclusión de fotografía

Al programar un componente visual (parte del módulo WAD de Openbravo), es necesario definir 3 elementos: la vista del componente, el controlador que responde a los eventos sobre ese componente, la forma en que el componente accede a los datos definidos en la base de datos. Aquí es evidente la existencia del patrón MVC, y estos componentes son la clave de su uso en la generación de ventanas automáticas Openbravo.

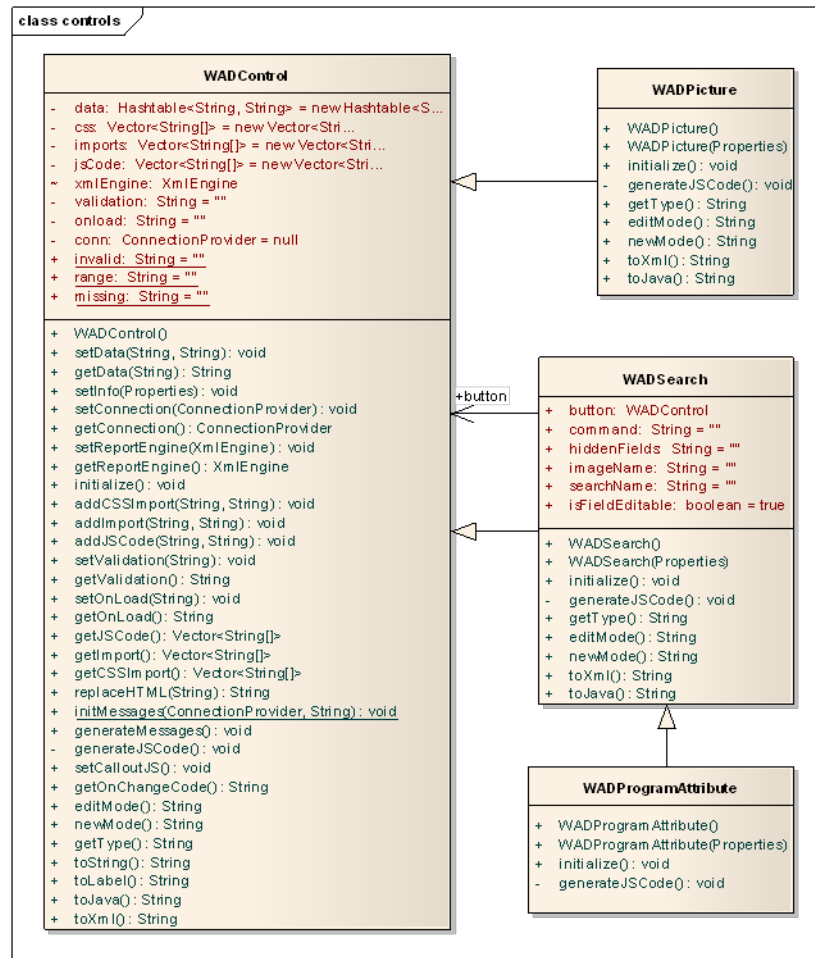


Ilustración 30: Diagrama de clases de los controladores WAD añadidos

En la Ilustración 30 se muestran la clase WADPicture que hereda de WADControl y es capaz de construir el componente visual capaz de visualizar la fotografía del usuario.

Por otro lado el controlador mostrado en la Ilustración 31 permite al usuario subir la fotografía del participante de dos formas. La primera es mediante la carga de un archivo de imagen existente en el sistema, la segunda es mediante la carga de una imagen capturada por una aplicación flash capaz de acceder a la webcam del equipo cliente. En ambos casos la imagen es redimensionada a un tamaño adecuado para su acceso y almacenaje.

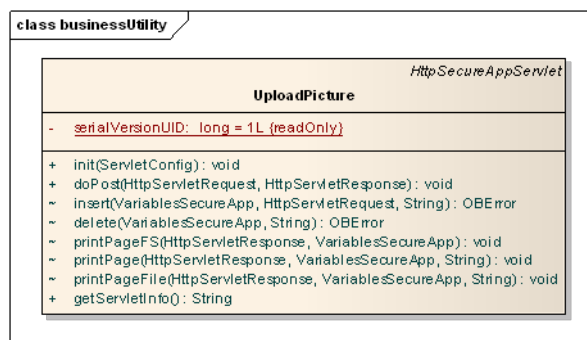


Ilustración 31: Controlador de carga de fotografías

1.3.4 Atributos Personalizados

El sistema Openbravo ERP, como parte de sus características ofrece un módulo de gestión de productos. Este módulo está construido sobre el módulo WAD de creación de ventanas automáticas, y entre sus componentes incluye uno que permite asignar a los productos parámetros personalizables.

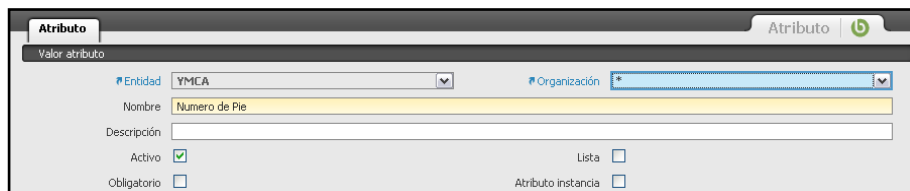
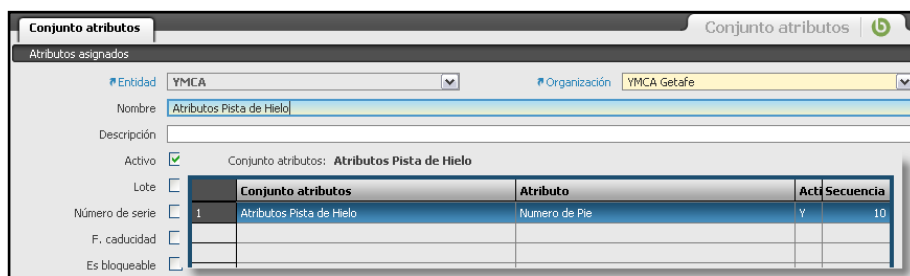


Ilustración 32: Creación de Atributo

Al definir los programas y actividades, se pensó en esta característica para poder añadir atributos extra como se hacía con los productos. Sin embargo, al ponerlo en práctica, resultó que ese componente estaba altamente acoplado a la gestión de productos y tuvo que ser modificado para permitir su utilización en programas y actividades.



Conjunto atributos	Atributo	Activo	Secuencia
1	Atributos Pista de Hielo	Numero de Pie	Y 10

Ilustración 33: Creación de Conjunto de Atributos

La aplicación permite definir atributos, conjuntos de estos atributos y asignar un determinado conjunto a un determinado producto. Una vez modificado el controlador, para permitir el uso en múltiples modelos, estos conjuntos pueden ser asignados también a programas y actividades.

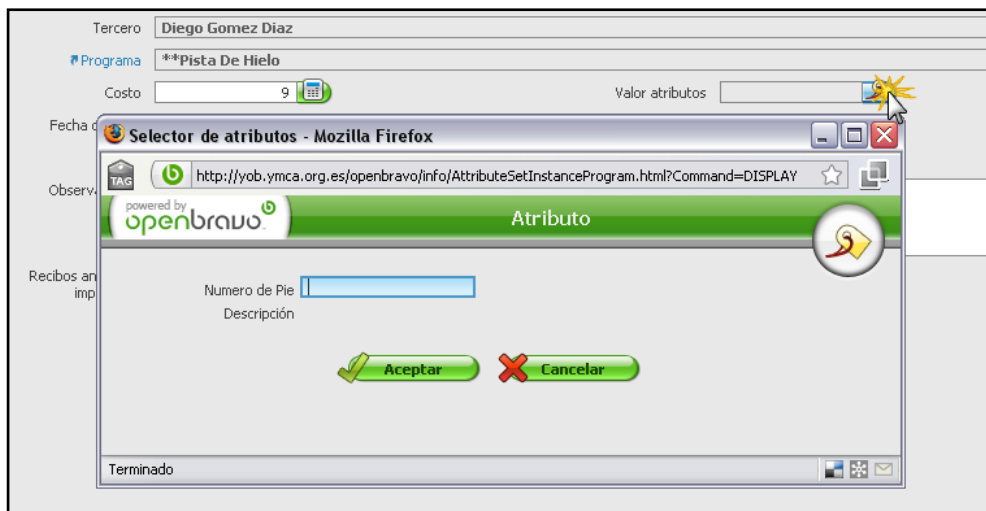


Ilustración 34: Edición de Atributo en Inscripción de Participante

Durante la creación de un programa o actividad, hay un campo que permite asignar el conjunto de atributo, que será posteriormente utilizado en la inscripción de un participante.

En la Ilustración 30 se muestra la clase `WADProgramAttribute` utilizada para implementar la gestión de atributos en programas y actividades. Esta clase se creó a partir de `WADAttribute`, capaz de gestionar sólo atributos en los productos.

1.4 Importación de los datos existentes

El origen de los datos con los que va a ser cargado el sistema en su puesta en funcionamiento, no es homogéneo. El análisis de la problemática de la asociación ya anticipaba que el proceso de migración de la información no iba a ser fácil. En los siguientes apartados se analiza los elementos a tener en cuenta y cómo realizar este proceso.

1.4.1 Análisis de los sistemas heterogéneos

Los datos se encuentran inicialmente en 4 fuentes de datos diferentes, 3 bases de datos Access y una hoja de cálculo Excel, ya que inicialmente se cuenta tan sólo con el conjunto de datos del área de Madrid.

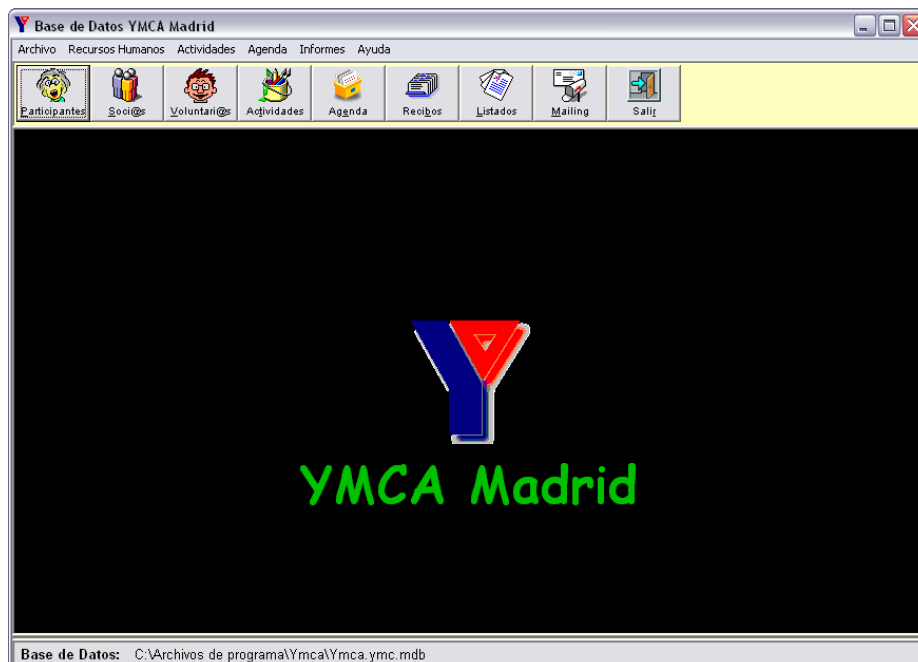


Ilustración 35: Sistema existente desde el que se quiere migrar

En el resto de sedes el panorama no es mucho mejor, ya que en el mejor de los casos cada delegación funciona con una hoja de cálculo o pequeñas bases de datos hechas en Access donde llevan el control de sus participantes. Algunas sedes incluso funcionan con registros en papel, y para estos la solución más rápida será introducir posteriormente la información de sus participantes a mano dentro del sistema.

La base de datos Access era utilizada por una aplicación antigua (Ilustración 35) escrita en Visual Basic (de la que no se conserva el código fuente). El modelo de datos de estas bases de datos se muestra en la Ilustración 36.

Aunque el modelo establece limitaciones en el número de bytes para los campos los datos tenían múltiples corrupciones que hacía que se superasen estos valores. Así aunque este modelo es orientativo la migración de los datos impondría limpiar los datos.

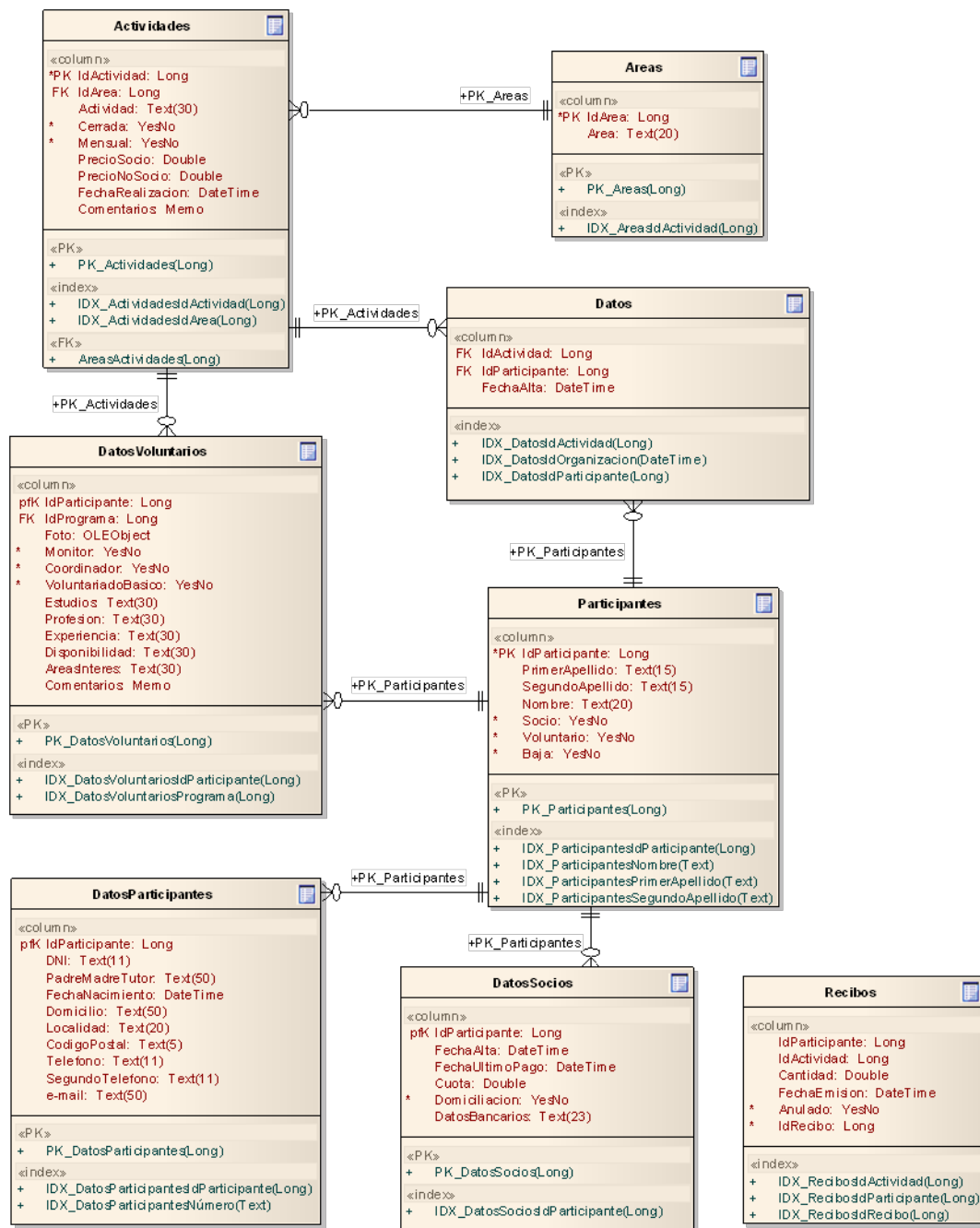


Ilustración 36: Modelo de datos base datos Access

El documento Excel existente contenía información básica de contacto: Nombre, Apellidos, Fecha de nacimiento, dirección postal, teléfono y nombre de tutor.

1.4.2 Definición de la correlación de datos

En el sistema nuevo existe cada modelo de datos del sistema original, y con bastante más información, así que el mapeo es bastante directo, pero se hacen necesarias algunas transformaciones:

- En el sistema destino los apellidos del usuario aparecen en un único campo, que en sistema original está dividido en primer y segundo apellido.
- Los campos booleanos en el sistema original son almacenados con el valor booleano TRUE y FALSE, pero en el destino son almacenados como 'N' (para FALSE) y 'Y' (para TRUE).
- Los identificadores de cada clave ajena es necesario convertirlo, asociando un valor temporal que posteriormente será reemplazado con su valor correcto.

1.4.3 Unificación de fuentes de datos

Al existir tres bases de datos diferentes de tres entidades que han colaborado entre sí y han compartido participantes y actividades es un hecho que las bases de datos contienen datos duplicados y con información más o menos actualizada.

Para unificar los datos del sistema habría que seguir los siguientes pasos:

- Insertar los datos en una base de datos temporal
- Comparar los duplicados a través de la limpieza de las cadenas (temporal), eliminando la acentuación y las mayúsculas.
- Ante dos coincidencias es coger los datos más actuales, o mediante inspección visual.

En cualquier caso, aunque el proceso de unificación se realice, es muy probable que persistan duplicados, por las faltas de ortografía cometidas en los nombres.

1.4.4 Creación del proceso de Migración

El proceso de migración es una combinación de la ejecución de instrucciones manuales y de scripts. En el apartado 3.2.6 de este capítulo se describe cómo se realizó el proceso, y más en detalle en Anexo II.

2 Desarrollo del Aplicativo

A continuación se describe la metodología y fases del desarrollo seguidas en el proyecto.

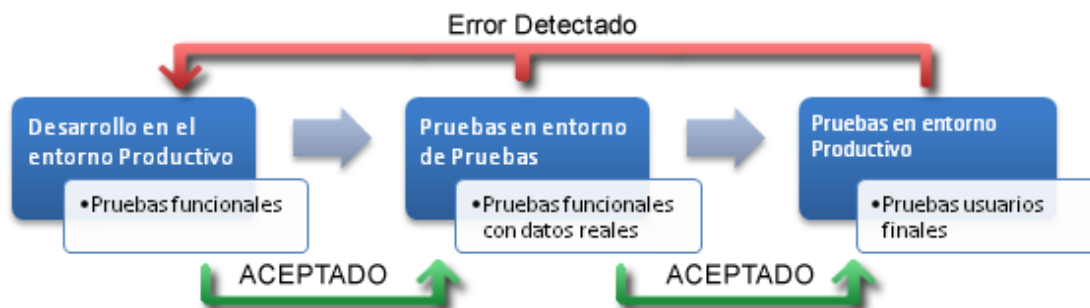


Ilustración 37: Metodología de desarrollo

2.1 Desarrollo en entorno de desarrollo

La implementación de cada módulo se realizó en el entorno de desarrollo, configurado para poder realizar pruebas rápidamente de los módulos creados gracias a la utilización de la aplicación ant, y al entorno de depuración de errores de Eclipse (12).

Cada módulo fue desarrollado según las especificaciones y las limitaciones que imponía la plataforma. Puesto que cada desarrollo requería la implementación del modelo de datos, seguido de su declaración en el sistema (configuración) y modificación de los controladores, tanto la implementación como el rediseño del código y del modelo de datos se hacía en paralelo.

Una vez completado un módulo se documentaba y traducía la interfaz, para hacerlo disponible tanto en Inglés (idioma base) como en castellano.

2.2 Pruebas en entorno de test

El entorno de pruebas fue introducido tras la instalación del sistema en el entorno productivo. Con el objetivo de replicar lo mejor posible el entorno productivo, pero sin el inconveniente futuro de dejar en un estado inaccesible el entorno productivo, como consecuencia de solucionar algún error.

2.3 Pruebas en entorno productivo

Las pruebas en el entorno productivo fueron el primer paso tras la implantación del sistema en este entorno, con el objetivo de validar funcionalmente cada característica desarrollada.

Sin embargo la idea era que los usuarios finales pudiesen validarlo por sí mismos, y al detectar un fallo nuevo, poder notificarlo para su corrección.

2.4 Corrección de errores

Una vez detectado y documentado un error, este debía ser reproducido, y resuelto pasando por las fases anteriores: corrección en entorno de desarrollo, prueba en el entorno de test, instalación en el entorno productivo con ayuda del sistema de control de versiones.

3 Implantación del sistema

La correcta implantación del aplicativo es una parte fundamental del proyecto, ya que se debe asegurar su correcta instalación y migración de datos que permitan ser utilizado inmediatamente tras ser completado.

3.1 Fases de implantación

Durante la etapa de implantación del producto se desarrollaron una serie de fases para instalar la primera versión del sistema, detectar y resolver los problemas que surgiesen.

- Instalación del sistema de control de versiones
- Instalación del sistema en el entorno productivo
- Instalación de la herramienta de control de incidencias
- Instalación del entorno de test
- Importación de datos

Algunos de las fases pueden no entenderse dentro del desarrollo de un proyecto como su orden natural, puesto que por ejemplo el repositorio de control de versiones se instala al comienzo de desarrollo y el entorno de pruebas antes que el productivo.

Sin embargo en este proyecto el despliegue de estos entornos se hizo según se creyó necesario y teniendo en cuenta que tan sólo hubo un desarrollador durante todo el proyecto que fue el autor, pero varios “testers” potenciales que eran los usuarios finales.

3.2 Procedimiento

A continuación se detalla el procedimiento que se siguió en cada una de las fases de la etapa de implantación.

3.2.1 Creación de repositorio de control de versiones

El objetivo de instalar un sistema de control de versiones a estas alturas del proyecto fue la de proporcionar un método eficaz de distribuir los futuros cambios que surgiesen como motivo de la corrección de problemas en la aplicación o la creación de nuevas funcionalidades, que serían realizadas en el entorno de desarrollo y desplegadas a un entorno de pruebas y al servidor productivo.

De esta forma, cada vez que se modificase algo en la aplicación, bastaba con notificarlo al control de versiones y en poco tiempo podrían estar replicados en el resto de entornos.

El sistema de control de versiones escogido fue Subversion(13) puesto que es el que originalmente se usaba en el proyecto Openbravo.

3.2.2 Instalación del sistema en entorno productivo

El sistema productivo era un servidor dedicado en el que residía la página web organizativa de la asociación, se optó por usar este entorno puesto que tenía recursos de sobra y no se esperaban grandes cargas concurrentes ni del servidor web ni de la aplicación ERP.

El sistema operativo de este servidor era Fedora Linux (14), que era diferente al usado durante el desarrollo y pruebas del sistema, que era Ubuntu (15). Esto trajo algunos problemas originados principalmente por las versiones que incluían ambas distribuciones de la máquina virtual de Java, el servidor de aplicaciones Tomcat(16) y la base de datos PostgreSQL (17).

Una vez preparado el entorno de instalación se realizó una copia de todo el código en el servidor para realizar una compilación y generación del archivo de despliegue para instalarlo en Tomcat.

La base de datos se crearía durante esta primera instalación a partir de los archivos xml que definen la estructura de la base de datos y que habían sido generados a partir del entorno de desarrollo.

Durante la instalación de la base de datos surgieron nuevos problemas, ya que la versión instalada de la base de datos, y su configuración interna, no permitían nombres de tablas, funciones, disparadores o registros en las bases de datos superiores a los 30 caracteres, restricción que era desconocida durante el diseño e implementación del sistema. Este problema hizo necesario renombrar los nombres y revisar los disparadores y funciones que los referenciaban, además del código de los diferentes componentes, y las entradas en el diccionario de la aplicación.

Una vez solucionado el problema se terminó de configurar el sistema, introduciendo los datos de la organización y definiendo los roles básicos de usuario a partir de la definición de tipos de usuarios que se recoge en la Ilustración 38, y que se determinó durante el análisis del sistema.

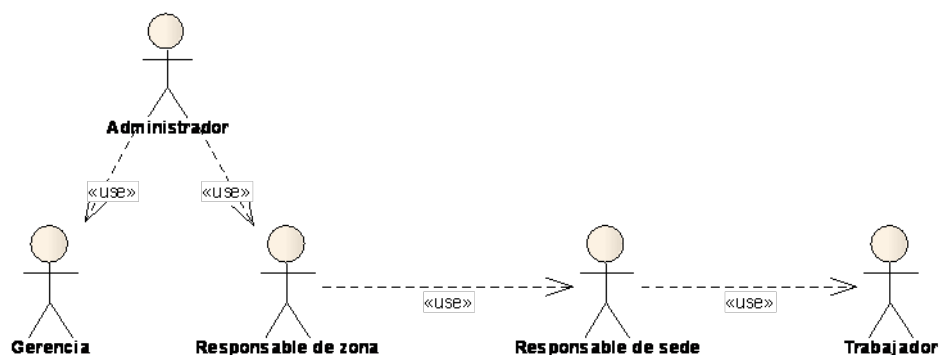


Ilustración 38: Roles de usuario

Sin embargo la utilidad del usuario Gerencia no será requerida hasta que se incorpore al sistema la funcionalidad financiera ofrecida por el ERP, tema que queda fuera del alcance del proyecto. Así, tanto los roles definidos en el sistema, como la funcionalidad que se concede a cada uno de ellos queda recogida en la Tabla 2.

Tabla 2: Roles de Usuario (X Acceso concedido - O Acceso parcial al módulo)

	Administrador	Usuario Avanzado	Responsable Sede	Trabajador Sede
Diccionario de la aplicación	X			
Configuración general	X	X		
Gestión de datos Maestros	X	X		
Gestión de compras	X			
Gestión de almacén	X			
Gestión de producción	X			
Gestión MRP	X			
Gestión de Ventas	X			
Gestión de Proyectos y Servicios	X			
Gestión de Programas	O	X	O	O
* Operaciones		X	X	O
** Gente		X	X	X
** Programas		X	X	X
** Proyectos		X	X	
** Recibos		X	X	X
* Informes		X	X	O
** Test Program Reports		X	X	
** List of Participants		X	X	X
* Configuración		X	X	
** Área de interés		X	X	
** Grupos de terceros		X	X	
** Atributo		X	X	
** Conjunto de atributos		X	X	
** Configurar Área		X	X	
Comunicación	O	X	O	O
* Mailing		X	X	X
* Mandar SMS		X	X	X
* Configuración		X	O	
** Servicios SMS		X		
** Cuenta SMS		X	X	
Gestión Financiera	X			
Web YMCA	X	X	X	X
Informe de error	X	X	X	X
Información	X	X	X	

Los usuarios de la aplicación no se crearon hasta que se realizó el curso formativo y se abrió oficialmente la plataforma a los trabajadores.

3.2.3 Configuración del sistema de backups

El sistema de la organización es un servidor dedicado ofertado por una empresa de hosting que incluye además en el servicio un servidor de backups disponible para esta necesidad.

Dentro del sistema YMCA Openbravo es necesario realizar copias de seguridad de dos tipos de datos: La base de datos del sistema y los archivos almacenados, tales como fotografías de usuarios y archivos adjuntos a registros.

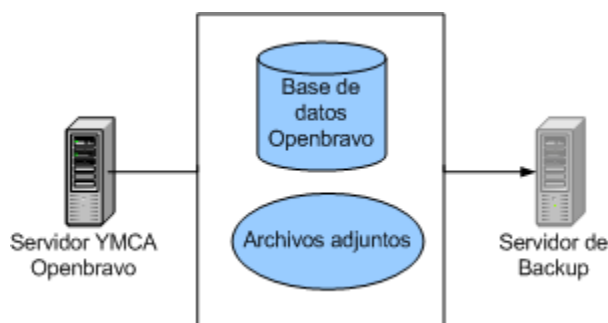


Ilustración 39: Backup del sistema

Para automatizar el proceso de backup se ha creado un script que será ejecutado por el demonio del sistema cron, cada noche a las 2 de la madrugada (fuera de la franja de disponibilidad exigida).

```
#!/bin/bash
PGPASSFILE=/root/.pgpass

# Nombre del archivo con día del mes (así cada mes se reciclan las copias)
filename=openbravoDump`date +%d'`.backup
filenameArchivos=archivos`date +%d'`.tgz

# Copia de la base de datos
/opt/PostgreSQL/8.3/bin/pg_dump -h localhost -U tad -d openbravo -F c -b -v -f
/tmp/$filename
bzip2 /tmp/$filename

# Copia archivos
tar czvf /tmp/$filenameArchivos /var/ymca-openbravo/attachments

# Configuración del servidor ftp
HOST=<Host ftp>
FTPUSER=<Usuario>
FTPPASS=<Password>

cd /tmp
ftp -n $HOST <<END
quote USER $FTPUSER
quote PASS $FTPPASS
cd openbravo
cd database
put $filename.bz2
cd ..
cd archivos
put $filenameArchivos
quit
END

rm -f /tmp/$filename*
rm -f /tmp/$filenameArchivos
```

3.2.4 Instalación del sistema de gestión de incidencias

Una vez desplegado el sistema en el entorno productivo se detectaron multitud de comportamientos anómalos, debidos fundamentalmente a dos factores. En primer lugar, el entorno de desarrollo era ligeramente diferente al productivo, tanto por las versiones del sistema operativo, como por la plataforma, ya que la del servidor era de 64bits, cambiando los binarios y las versiones, tanto de la base de datos, como del servidor de aplicaciones.

Por otro lado, y más importante, al realizar la instalación limpia en el entorno productivo, se creó una nueva empresa con una configuración que hubo que crear desde cero. Los módulos desarrollados no tuvieron en cuenta los cambios de una empresa a otra, donde el identificador de la base de datos resultó ser más importante de lo esperado.

Puesto que las pruebas se fueron realizando validando funcionalmente cada módulo, y sin establecer un proceso formal, no había forma de asegurar que el sistema funcionase correctamente, sin la ayuda de los usuarios finales, cuya forma de trabajo suele ser muy diferente a la del desarrollador, y de ahí su gran mayoría de comportamientos impredecibles.

Con esta idea en mente, se instaló una plataforma que permitiese a los usuarios reportar los fallos que encontrase y a la vez pudiesen hacer un seguimiento de su completitud. Se optó por esta solución más que por una atención personalizada, a priori, para llevar un registro de los errores resueltos, es establecer un mecanismo común para los trabajadores y proporcionar un medio a través del cual solicitar nuevas funcionalidades.

Las nuevas funcionalidades que se prevén son sobre todo la generación de nuevos informes, con información útil que almacena el sistema, y que los trabajadores pueden necesitar en su día a día, pero que no se detectaron.

3.2.5 Instalación del entorno de test

Dada la diferencia entre los entornos de desarrollo y de producción se optó por utilizar un entorno intermedio, como se hace en los proyectos empresariales, que fuese lo más parecido al entorno productivo y se pudiesen probar el sistema modificado antes de trasladarlos definitivamente al entorno productivo.

El entorno de pruebas se instaló en una máquina virtual para permitir hacer todo tipo de pruebas y poder recuperar la máquina en su estado original cuando fuese necesario. Debía tener las mismas versiones del gestor de bases de datos, del servidor de aplicaciones y de la implementación de Java.

Además, para asegurar que la aplicación se comportase lo más fielmente posible, antes de realizar las pruebas se hacía una copia limpia de la base de datos del entorno productivo y probaba el código recién modificado.

3.2.6 Importación de datos

El proceso de importación es uno de los más importantes y laborioso del proceso de implantación. En esta fase se han de tomar los sistemas heterogéneos que existen estudiarlos y extraer de ellos la información de los participantes y programas existentes.

Aunque la aplicación Openbravo permite realizar el proceso de importación a partir de un origen de datos en CSV, hay algunos datos que no contemplaba la aplicación original y para el módulo de importación funcionase correctamente habría que modificarlo. Así se optó por hacer una inserción directa en la base de datos, con ayuda de nuevo con la herramienta Talend Studio.

Puesto que la información es gestionada en diferentes sistemas, hay información replicada y con formatos muy diferentes. Como formato de salida se utilizará un listado de participantes con sus datos de contacto, en CSV (texto plano separado por comas), que puede ser importado por la aplicación Openbravo.

Para realizar todo este proceso se utiliza la herramienta Talend Studio (18), que permite crear los scripts de importación de forma visual, a partir de las herramientas que ofrece.

A continuación se describen los pasos llevados a cabo:

Unificación de los datos procedentes de la base de datos Access y las hojas de cálculo Excel.

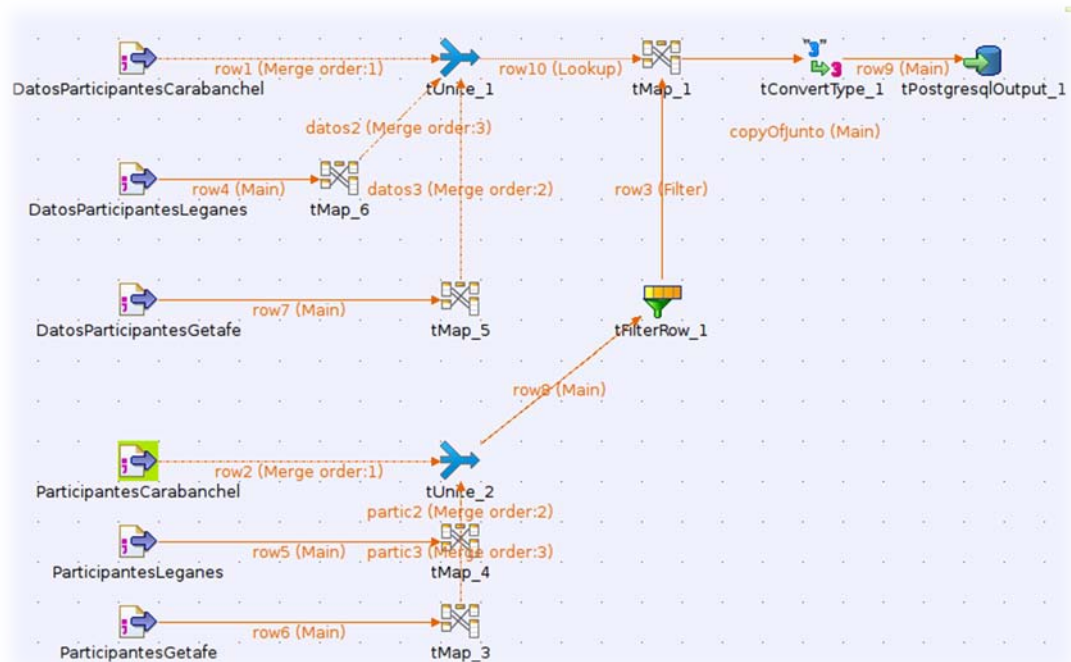


Ilustración 40: Importar participantes desde las tres fuentes de datos existentes

Puesto que es de este tipo de datos no se puede gestionar directamente en Linux sin ayuda de algún driver propietario, que no funcionan del todo bien, se convirtieron las diferentes tablas de la base de datos al formato CSV. De esta forma se puede acceder directamente a los datos sin depender de la base de datos.

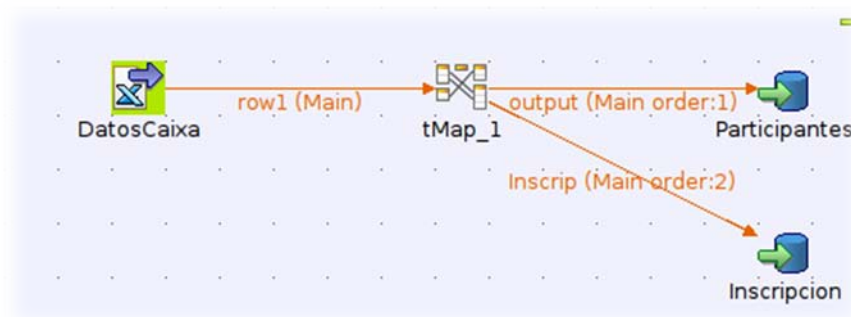


Ilustración 41: Importación desde archivo Excel

Una vez obtenidos estos datos se unificaron la lista de participantes siguiendo los siguientes criterios:

- Usuarios marcados como activos en la base de datos.
- Cuyo apellido no comenzase por "." (Procedimiento alternativo usado por responsables de la asociación para identificar usuarios no activos).
- Conservar una sola copia de cada registro, pero guardando su identificación para conservar las relaciones en otras tablas.

Estos datos se migraron a una base de datos temporal PostgreSQL para realizar filtros más elaborados posteriormente.

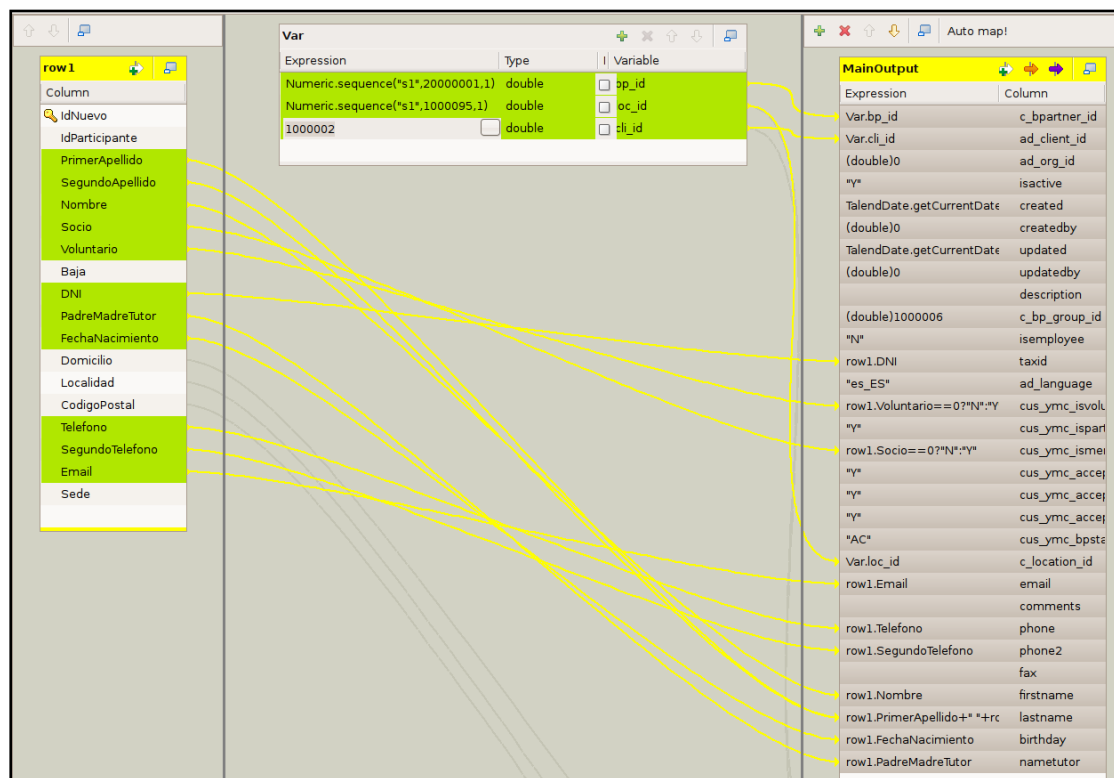


Ilustración 42: Mapeo de datos en Talend Studio

A esta base de datos temporal también se copiaron los usuarios almacenados en un documento Excel (otro de los procedimientos usados en la organización). Evitando replicar datos.

De las bases de datos existentes también se recuperaron las referencias a inscripciones para posteriormente replicar estas inscripciones en el nuevo sistema.

En el Anexo II se recoge en detalle los pasos seguidos y las instrucciones SQL ejecutadas para realizar algunas de las conversiones.

4 Programa formativo

Un sistema si no es conocido está destinado al fracaso. Ante esta premisa, se requiere que el sistema sea bien conocido por los miembros de la entidad para que sea exitosa su implantación.

4.1 Antecedentes

La entidad instauró hace unos años el sistema de aprendizaje online Moodle(19) dentro de sus escuelas formativas de ocio y tiempo libre. Este sistema se ha utilizado hasta ahora como un repositorio de documentación que producian los docentes, pero nunca se realizó un diseño de curso online en condiciones.

Este tipo de cursos ofrecen la ventaja de pueden cursarse en cualquier momento y lugar, donde el ritmo de aprendizaje está marcado por el propio estudiante. Sin embargo la mayoría de estos cursos son monótonos y carecen del estímulo que un docente puede inyectar en un curso tradicional.

Por mi experiencia en este tipo de cursos y la que obtuve en la asignatura “Enseñanza asistida por ordenador”, puedo afirmar que este tipo de cursos sólo triunfan si ofrecen una variedad adecuada que sea concorde a la etapa del aprendizaje.

Diseñar un buen curso es una tarea laboriosa y que lleva tiempo, sin embargo un buen diseño hace mucho más amena su participación y esto era un pre-requisito obligado, para animar a los técnicos, generalmente muy reacios, a cursarlo con éxito.

4.2 Diseño de curso SCORM

El curso ha sido diseñado siguiendo el estándar SCORM propuesto por ADL (20). Este estándar se está convirtiendo en uno de los más utilizados actualmente, y que mejor está soportado por las diferentes herramientas de aprendizaje.

El estándar SCORM proporciona un mecanismo de empaquetado de los contenidos educativos, además de establecer la jerarquía de los diferentes contenidos, y mecanismos para hacer el seguimiento de los estudiantes, que hayan completado correctamente cada unidad. La definición de todas sus características se realiza por medio de un archivo XML.

Como herramienta de edición se utilizó eLearning XHTML editor (21). Este editor permite editar los recursos educativos añadiendo diferentes componentes a las páginas web, y posteriormente exportarlos en varios formatos. Entre los formatos disponibles se encuentra el paquete SCORM.

El curso generado incluye el índice de contenidos del manual de usuario (disponible en el Anexo I), además incluye videos que describen en detalle algunos de los procesos. En la Ilustración 43 puede verse una captura del curso desplegado en la plataforma de aprendizaje Moodle de la organización.

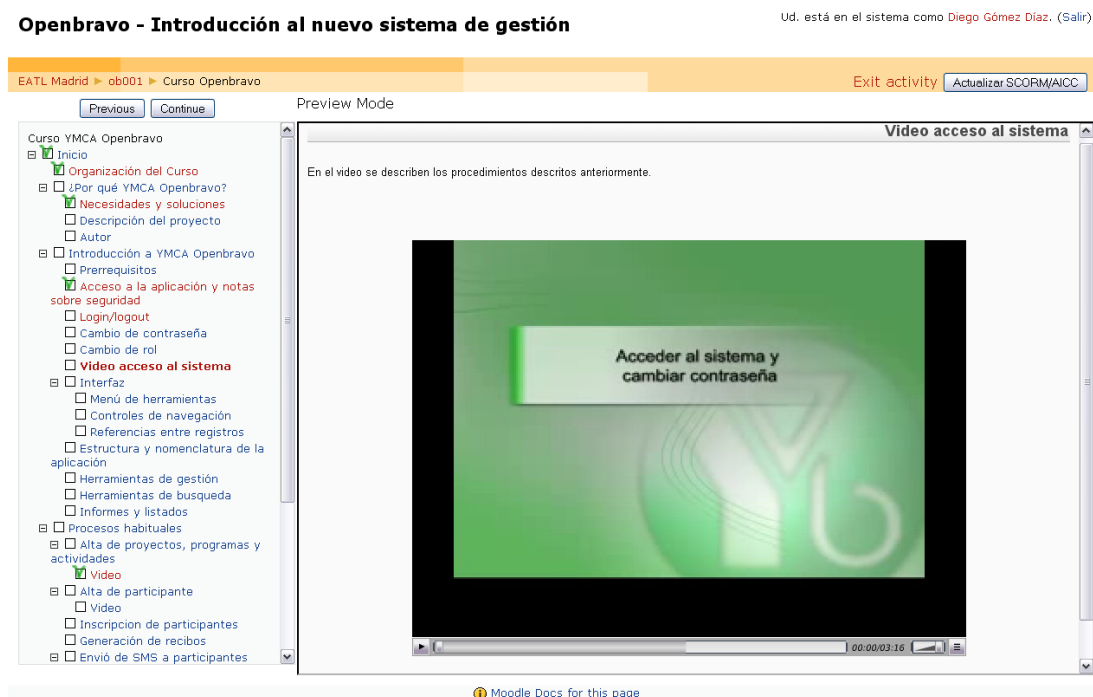


Ilustración 43: Curso SCORM desplegado en la plataforma moodle

En el DVD que acompaña al proyecto se ha añadido el paquete SCORM generado y un directorio que contiene el curso disponible para visualizar desde el navegador sin la necesidad de una plataforma de aprendizaje.

4.3 Curso formativo presencial

Como parte del programa formativo del proyecto, se realizaría un curso formativo presencial dirigido a los técnicos de la asociación para presentarles la plataforma y la funcionalidad básica del sistema. Este curso pretendía complementarse con el curso online que ofrece la información mucho más detallada.

Sin embargo, la ventaja de un curso presencial, es que el docente puede evaluar mucho mejor el nivel de aprendizaje adquirido por la percepción visual y el nivel de atención prestada durante el curso.

Al arranque del proyecto era importante realizar un curso formativo presencial para garantizar que los primeros usuarios del sistema comprendiesen que aportaba el nuevo sistema y cómo podían comenzar a utilizarlo.

Durante este curso se entregaría el manual de usuario que se recoge en el Anexo I.

Planificación

En este apartado se recogen algunos comentarios acerca de la planificación del proyecto, las variaciones sufridas desde el comienzo hasta el final, y el diagrama con la planificación detallada.

En total la duración del proyecto, incluyendo la elaboración de la propuesta y el pre proyecto supera el año de duración, con periodos de actividad muy variados.

1 Planificación Inicial

El proyecto en su arranque planteó varias líneas de trabajo que habrían variado la planificación del proyecto.

De cara al proyecto en la asociación, se pretendía inicialmente cubrir mayor funcionalidad de cara a la inclusión de la función financiera. Esta característica es tá implementada en Openbravo, y la línea de desarrollo habría ido por adaptar la contabilidad a las imposiciones legales de las organizaciones no lucrativas, y a asociar los gastos e ingresos de los programas con la contabilidad. Sin embargo esta funcionalidad no fue incluida al final porque el proceso que sigue la asociación a nivel contable y financiero no se va a cambiar a corto plazo. Así la herramienta pasaba a ser un complemento más que un sustituto.

Por otro lado, vistas las posibilidades del proyecto y las oportunidades que podía ofrecer este a otras entidades similares, se planteó la posibilidad de crear un producto de software configurable para que fuese adaptable por otras entidades. Sin embargo esto imponía mayor tiempo de desarrollo y un compromiso mayor de cara a otras entidades.

2 Planificación Final

En el diagrama de Gantt mostrado en la Ilustración 44 se muestra en detalle las principales tareas realizadas en el proyecto y su duración. El tiempo que abarca el diagrama va desde el 1 de Mayo de 2008 hasta el 20 de Mayo de 2009, sin incluir el tiempo dedicado a la elaboración de la memoria del Proyecto.



Ilustración 44: Cronograma del Proyecto

En este periodo una de las tareas más largas e intensas fue el proceso auto formativo de la plataforma. Aunque hay bastante información disponible de la plataforma Openbravo en su página web y en otras, esta información es más útil a nivel funcional que

para los desarrolladores. Así el aprendizaje de parte de la plataforma y de la iteración entre los diferentes elementos se tuvo que llevar a cabo observando el código fuente.

El desarrollo tampoco fue fácil, ya que implementar algunos procesos no era tan intuitivo como parecía inicialmente, esto produjo que la resolución de errores y la reimplantación de algunos módulos implicasen un aumento del tiempo de desarrollo.

En la última etapa del proyecto, el tiempo que se pudo dedicar no fue mucho debido a que el autor tuvo que compatibilizar las clases en la universidad, con el proyecto y una beca de 7 horas diarias. Esto motivó que el proceso de migración de datos se completase en un periodo tan largo de tiempo.

3 Presupuesto

Con el objetivo de tener una estimación acerca del coste aproximado del proyecto, se proporciona el siguiente presupuesto basado en las horas invertidas en el proyecto y las licencias de software necesarias para su desarrollo.

3.1 Licencias de Software

Uno de los objetivos de este proyecto era minimizar los costes de adquisición del sistema a través de la utilización de software de licencia libre. En la Tabla 3 muestra la relación de las aplicaciones de software utilizadas para realizar el proyecto.

Tabla 3: Coste de Licencias de Software

Software	Coste de licencia
Apache Tomcat (16)	0 €
Eclipse Framework (12)	0 €
eXe eXeLearning (21)	0 €
Fedora Linux (14)	0 €
Jasper iReport (7)	0 €
Microsoft Office 2007 (Student versión)	700 €
Microsoft Project	600 €
Microsoft Windows Vista	130 €
Openbravo ERP(3)	0 €
Postgres SQL (17)	0 €
Talend Studio (18)	0 €
Tigris Subversion (13)	0 €
Total	1.430 €

3.2 Recursos humanos

Las tareas realizadas durante el proyecto corresponden a diferentes roles de diferentes costes. Asumiendo que la media invertida de horas invertidas en el proyecto por día ha sido de entre 3 y 4 horas, durante la duración del proyecto detallado en diagrama de Gantt de la Ilustración 44, la relación total de horas invertidas por rol y su coste queda reflejado en la Tabla 4.

Tabla 4: Coste de recursos humanos

Rol	Coste/hora	Horas	Coste Total
Jefe de Proyecto	60€	220	13.200€
Analista	40€	550	22.000€
Programador	30€	525	15.750€
Test	20€	170	3.400€
Documentación	30€	205	6.150€
Total			60.500€

3.3 Coste de hardware

Teniendo en cuenta las necesidades de hardware para el desarrollo y la implantación del sistema (sin incluir los equipos clientes). Los costes que incluyan un plan de mantenimiento durante al menos dos años quedan reflejados en la

Tabla 5: Coste hardware

Sistema	Descripción	Coste
Desarrollo y Test	Sistema utilizado para el desarrollo del sistema	600€
Servidor de producción	Plan de hosting en un servidor dedicado en el proveedor 1and1, durante 2 años de contrato.	1.680€
Total		2.280€

3.4 Balance final del presupuesto

Finalmente se proporciona un presupuesto final (€) del que hay que destacar que se ha incluido una línea presupuestaria en concepto de riesgo, que habría que tener en cuenta en caso de un proyecto nuevo y que en caso de un proyecto empresarial habría que tener en cuenta además el beneficio esperado, que no ha sido incluido para simplificar el balance.

Tabla 6: Presupuesto del proyecto

Concepto	Coste Total
Recursos Humanos	60.500€
Licencias de Software	1.430€
Hardware	2.280€
Riesgo (10%)	6.400€
Total sin IVA	70.610€
IVA (16%)	11.297€
Coste Final	81.907€

Resultados

1 Resultados alcanzados

La implementación del proyecto se dio por finalizada el 11 de Enero del 2008, tras esta fecha comenzó el proceso de implantación que se realizó inicialmente en 3 de las sedes de la asociación.

Tras la implantación inicial, debido a la falta de tiempo suficiente, y lo largo que fue el proceso de migración de datos, el sistema no entro en funcionamiento hasta mediados del mes de Mayo.

A mediados del mes de Abril, se realizó con éxito una migración de los datos existentes de las aplicaciones anteriores, al nuevo sistema.

A finales del mes de Abril se impartió un curso formativo del sistema (recogido en el Anexo I) a los técnicos de la asociación. Este hito supuso el arranque oficial del sistema, y comenzó a ser utilizado coincidiendo con el arranque de la campaña de verano.

En los primeros días de la campaña de verano, donde la aplicación podía mostrar todo su potencial, al mejorar el trabajo en red que hasta el momento había, la aplicación comenzó a usarse en paralelo con la anterior base de datos. Sin embargo, debido a uno de los habituales errores del anterior sistema, que se volvía inestable al realizar múltiples accesos, dejó de utilizarse el antiguo para usarse en exclusiva el nuevo.

Este hecho ha permitido experimentar el comportamiento real de la aplicación e ir corrigiendo errores de funcionamiento que no se habían detectado.

Dada la delicadeza de la época de inscripción de campamentos, en estos días el soporte que se dio fue continuo para asegurar que los técnicos podían trabajar sin dificultades causadas por la aplicación. No obstante, los incidentes reportados no fueron ni muy numerosos, ni muy graves, y los flujos de trabajo de la aplicación fueron aprendidos en poco tiempo.

La evaluación por parte de los trabajadores, a corto plazo es muy positiva, ya que ha mejorado el intercambio de información entre los compañeros de diferentes sedes y la curva de aprendizaje no ha sido muy elevada.

De cara a la integración en el resto de sedes, no se ha podido realizar aún, porque la gerencia no está segura de si dar el paso o no, preocupados por la falta de experiencia de las otras sedes en este tipo de sistemas.

2 Proyectos similares

En paralelo, a la vez que se producía este proyecto, dos iniciativas han surgido para ofrecer soluciones a ONGs bajo la misma plataforma Openbravo.

La primera es una realidad ya distribuable y dirigida a entidades no lucrativas que trabajen para personas con discapacidad. Este proyecto recibió la denominación de Gesmon(11) y el director del proyecto Ángel Zambrana Fernández. Por conversaciones mantenidas con el responsable, se supo que este proyecto había sido encargado directamente a Openbravo.

La segunda es un encargo de un proyecto similar al del proyecto que habría sido encargado por la Fundación Luis Vives también a Openbravo. Esta fundación trabaja por mejorar la productividad en las entidades no lucrativas proveyéndole de medios para su trabajo. Es probable que el encargo desemboque en un producto abierto para que lo utilicen las entidades que lo soliciten.

Analizando estos proyectos la apuesta realizada por esta plataforma en su momento para realizar el proyecto se ve más que afianzada y en el futuro puede verse reforzado el proyecto por aportaciones de estos proyectos.

3 Líneas futuras de trabajo

El proyecto Openbravo es uno de los proyectos Opensource más vivos de la comunidad y quiere seguir posicionándose a través de mejorar la arquitectura y facilitar la colaboración de terceros.

En la última etapa del proyecto, se liberó la siguiente versión del producto Openbravo ERP (2.50). Entre sus características más destacadas está inclusión de un gestor de componentes, que permite instalar y desinstalar componentes de terceros con una gran facilidad, comparando el proceso que se hubo de seguir en este proyecto, es un gran avance.

Además se ha incluido la gestión del modelo a través de Hibernate, para ganar en la independencia del gestor de bases de datos, y abstraer toda la complejidad mucho más de lo que ya venía haciendo con la definición de las consultas en XML.

En futuras versiones se pretende mejorar la experiencia del usuario, y la comunicación con sistemas externos.

Sería conveniente migrar los componentes realizados en este proyecto, para hacerlos compatibles con la nueva versión y aprovechar la nueva característica de gestión de componentes, para distribuir el trabajo hecho, para que pueda ser aprovechado. En la página del proyecto Openbravo se ofrece una documentación(22) completa acerca de los elementos que habría que tener en cuenta para realizar la migración.

De los sistemas similares que se han liberado o lo harán pronto sería interesante aprovechar algunos elementos, como la gestión de recursos humanos.

4 Conclusiones

A lo largo del documento se ha tratado de plasmar los elementos fundamentales del trabajo realizado, pero es importante comprender el alcance real de este proyecto en términos de su aplicación e influencia futura. El proyecto ha conseguido poner en funcionamiento un sistema de gestión en una ONG, y crear el punto de partida de un proyecto que con suerte sea continuado durante los próximos años e influya positivamente en la productividad de esta organización, para que los profesionales que en ella trabajan puedan dedicar su tiempo a lo que realmente saben hacer bien, el trabajo con sus destinatarios, y que los recursos disponibles sean gestionados del mejor modo posible.

La plataforma introducida es fácilmente ampliable y mantenible y la apuesta, que otras organizaciones parecidas y empresas de soporte están haciendo por este sistema, garantiza la vida del software.

Además se ha tratado de facilitar el camino de los trabajadores que harán uso de este sistema, proveyéndoles de una herramienta lo más parecida a la que tenían anteriormente (los que la tenían), y produciendo documentación que facilite su aprendizaje.

A nivel metodológico el proyecto ha incluido un proceso completo de estudio de la problemática y las alternativas, elaboración de propuesta, desarrollo e implantación de un producto de software. Se han seguido técnicas y procedimientos propios de las grandes empresas, interpretando roles diferentes, tratando de afrontar cada tarea de la mejor manera posible. Para esta aplicación se han seguido las recomendaciones ofrecidas por la comunidad que hay en torno a Openbravo, algunos de los procedimientos prácticos de las metodologías estudiadas en la carrera y las técnicas utilizadas en empresas como Deloitte, gracias a la experiencia adquirida al participar en un proyecto real de esta empresa.

Transversalmente se ha desarrollado todo un proceso formativo que ha ido desde conocer a fondo todos los elementos que influyen en la vida de un producto de software, desde la concepción y diseño de un producto de software a su liberación y mantenimiento. A la vez que participaba en un proyecto de software libre, estudiando la documentación disponible, generando nueva, detectando fallos y aportando nuevas funcionalidades. Ya que en el momento que se empezó a trabajar sobre Openbravo 2.40 (base del proyecto), este producto estaba en fase de pruebas.

Además la magnitud de este proyecto he permitido desarrollar técnicas y cualidades aprendidas en algunas de las siguientes asignaturas de la carrera de Ingeniería Informática:

- Ingeniería de Software: Por las técnicas y metodologías para la elaboración de proyectos de software, cubriendo las diferentes etapas de la vida de un producto de software.
- Diseño y administración de bases de datos: Establece las bases de la gestión y diseño de bases de datos, que son cruciales en este tipo de sistemas de gestión empresarial.
- Programación avanzada: Uso de técnicas y herramientas útiles en estos sistemas.
- Servidores de información: Configuración de servidores de aplicaciones y comprensión de los protocolos que intervienen en las comunicaciones.
- Enseñanza asistida por ordenador: Estándares y herramientas utilizados en los entornos de aprendizaje.

Sin duda, este proyecto ha sido muy completo y ha cubierto con creces los objetivos de la realización de un proyecto fin de carrera. La plataforma utilizada resulta muy interesante a nivel pedagógico, y la considero muy apta para desarrollar futuros proyectos fin de carrera, por parte de otros alumnos de la titulación.

Referencias

En esta sección recogemos las fuentes de algunas fuentes de información que no s han permitido realizar la práctica.

1. **YMCA España.** YMCA España. *Web Principal de la organización.* [En línea] <http://www.ymca.org.es>.
2. **SugarCRM Inc.** SugarCRM. *CRM de código abierto.* [En línea] <http://www.sugarcrm.com/crm/>.
3. **Openbravo.** Wep principal de Openbravo ERP. [En línea] <http://www.openbravo.org>.
4. **Apache.** Struts. *Free open-source framework for creating Java web applications.* [En línea] <http://struts.apache.org/>.
5. **SpringSource.** Spring Framework. [En línea] <http://www.springsource.org/>.
6. **Red Hat Middleware.** Hibernate. *Object/relational persistence and query service.* [En línea] <https://www.hibernate.org/>.
7. **Jasper.** iReport. *Editor visual de informes de la librería Jasper Report.* [En línea] <http://jasperforge.org/projects/ireport>.
8. **Openbravo.** Openbravo Community. *Wiki con documentación del proyecto.* [En línea] <http://wiki.openbravo.com/wiki/>.
9. **Cares Ramirez, Maximino.** *OPENBRAVO Guía del Usuario 2.3x.* Cuarta. s.l. : MIND Information Technologies SpA, 2008. 978-956-319-222-3.
10. **Telepieza.** Telepieza's Weblog. *Documentación y Tutoriales sobre Openbravo ERP.* [En línea] <http://www.telepieza.com/wordpress/category/informatica/opensource/openbravo/>.
11. **Zambrana Fernández, Ángel.** Gesmon. *Proyecto dirigido al tercer sector realizado sobre Openbravo.* [En línea] <http://www.gesmon.net/>.
12. **Eclipse.** Eclipse main website. *Opensource software development platform.* [En línea] <http://www.eclipse.org/>.
13. **Tigris.** Subversion - Open Source Version Control System. [En línea] <http://subversion.tigris.org/features.html>.
14. Fedora Operating System. [En línea] <http://fedoraproject.org/wiki/Overview>.
15. Ubuntu Operating System. [En línea] <http://www.ubuntu.com/>.

16. **Apache.** Tomcat. *Implementation of the Java Servlet and JavaServer Pages technologies*. [En línea] <http://tomcat.apache.org/>.
17. **PostgreSQL Global Development Group.** PostgreSQL - Open Source database. [En línea] <http://www.postgresql.org/>.
18. **Talend.** Talend: Open Source ETL and Data Integration Software. [En línea] <http://es.talend.com/index.php>.
19. Moodle - Learning Management System. [En línea] <http://moodle.org/>.
20. **Advanced Distributed Learning.** SCORM. [En línea] <http://www.adlnet.org/Technologies/scorm/default.aspx>.
21. **EXE.** eLearning XHTML editor. [En línea] <http://exelearning.org/>.
22. **Openbravo.** Migración a Openbravo 2.50 desde 2.40. [En línea] http://wiki.openbravo.com/wiki/ERP/2.50/Upgrading_From_2.40.

Anexo I – Manual de usuario

1 Organización del curso

El presente curso ha sido creado para proporcionar un primer contacto de los técnicos de YMCA a la futura aplicación de gestión de la entidad YMCA Openbravo.

1.1 *Objetivos*

El curso ha sido concebido para alcanzar los siguientes objetivos:

- Conocer los elementos principales de la herramienta YMCA Openbravo para ser capaces de usarla con soltura.
- Ser capaces de realizar las operaciones habituales en el día a día de la asociación.
- Conocer el procedimiento que se ha establecido para reportar errores y solicitar nuevas funcionalidades.
- Conocer y utilizar los medios se han establecido para ampliar el conocimiento de la aplicación.

1.2 *Ejercicios y material del curso*

Como parte del curso se realizará una serie de ejercicios que tienen como objetivo el asentar las descripciones realizadas de los procesos habituales.

Al comienzo de cada ejercicio se proporciona un objetivo y una descripción general de la tarea. Se recomienda realizar el ejercicio sin seguir al pie de la letra las descripciones, usando estas sólo como método orientativo si la intuición no nos orienta adecuadamente.

2 ¿Por qué YMCA Openbravo?

En esta sección se describe a modo informativo cuáles fueron los motivos de la creación del producto YMCA Openbravo y cómo ha sido concebido.

2.1 *Necesidades y Soluciones*

Para los trabajadores de la asociación son bien sabidas las limitaciones actuales en cuanto a la aplicación de las nuevas tecnologías en la gestión de las actividades y participantes.

Son frecuentes los errores de los sistemas actuales, y el proceso de compartir información es desordenado y muy lento.

La solución ofrecida pretende resolver en primer lugar estos problemas, ofreciendo una arquitectura apta para el trabajo en red, y pensada para poder ser cambiante en el futuro de acuerdo a las necesidades venidera.

2.2 *Descripción del proyecto*

YMCA Openbravo es el fruto de un proyecto fin de carrera cuya principal motivación era resolver la problemática con la que YMCA se encuentra de cara a la gestión informatizada de sus programas, participantes y sus implicaciones contables.

El proyecto arrancó en Mayo del año 2008, y a lo largo de 11 meses se ha dado forma para conseguir el producto actual.

2.3 *Autor*

El autor del proyecto (Diego Gómez Díaz) ha sido voluntario de la organización durante los últimos 6 años. Tiempo en el que ha participado en diferentes áreas de la asociación lo que le ha permitido adquirir un conocimiento profundo sobre las diferentes problemáticas y necesidades de la asociación.

Como parte de la formación académica el autor se especializó en Sistemas informáticos distribuidos, lo que le ha permitido proporcionar una solución muy adecuada a las necesidades de la asociación.

Durante la última etapa del proyecto, el autor estuvo trabajando para la empresa Deloitte, consultora de gran prestigio, en el área de certificación de aplicaciones, lo que le ha permitido aplicar procedimientos de calidad y aseguramiento del Software propio de grandes corporaciones.

3 Introducción a YMCA Openbravo

YMCA Openbravo es un Software de gestión construido sobre la plataforma Openbravo dedicada a la gestión contable de pequeñas y medianas empresas.

Como veremos más adelante al utilizar Openbravo como la base del producto hay ciertas herencias en los procedimientos de trabajo que hay que mantener.

3.1 Prerrequisitos

Para poder utilizar YMCA Openbravo es necesario contar con una cuenta de usuario, que es proporcionada al comienzo de este curso, y un navegador de los siguientes un ordenador con acceso a Internet:

- Mozilla Firefox 1.5¹ o posterior (Navegador recomendado)
- Internet Explorer 7.0 o posterior (Puede presentar algunos problemas)
- El resto de navegadores tienen dificultades para visualizar correctamente la interfaz y por tanto no se recomiendan.

La aplicación YMCA Openbravo ha sido instalada en el mismo servidor que la página web de la asociación, así que si puede acceder a la página web podrá acceder al servidor.



Se han detectado problemas al generar informes en PDF al tener activados ciertas extensiones en los navegadores Mozilla Firefox e Internet Explorer. Si experimenta estos problemas trate de desactivar los componentes, estos navegadores recién instalados es como mejor funcionan.

¹ <http://www.mozilla-europe.org/es/firefox/>

Para permitir que la aplicación funcione correctamente es necesario desactivar en el navegador el bloqueo de ventanas emergentes para el dominio yob.ymca.org.es.

Los informes y listados son generados en PDF por lo que es necesario tener instalado Adobe Acrobat Reader².

En las fichas de participantes es posible añadir una fotografía bien a través de un archivo o mediante la captura de una webcam, en este último caso es necesario disponer de una, sin embargo esto es completamente opcional.

3.2 Acceso a la aplicación y notas sobre seguridad

Para trabajar con la aplicación YMCA Openbravo es necesario disponer de una cuenta de usuario que se proporciona al comienzo del curso, esta cuenta dispone de una serie de permisos que le permite realizar determinadas funciones que son registradas en el sistema.

Cada día es más importante asegurar la correcta gestión de los datos privados de nuestros participantes, al implantar la nueva aplicación en internet frente al anterior acceso local debemos ser más conscientes de los posibles riesgos y por eso se mantienen dos medidas de seguridad fundamentales.

- Las operaciones de cada usuario son registradas, por lo que cada uno es responsable de cuidar sus credenciales y actuar de una forma responsable en el sistema.
- La contraseña como mecanismo principal de acceso debe cumplir con criterios adecuados de seguridad y mantenerse en secreto.

Una contraseña fácil de averiguar no tiene por qué ser difícil de recordar. Sin embargo una contraseña basada en palabras, nombres, fechas o secuencias de teclado sencillas son las primeras en ser descubiertas por un atacante malintencionado.

Cuidemos de la información confidencial de nuestros participantes.

² <http://www.adobe.com/es/products/reader/>

3.3 Login/Logout

Para poder acceder a la página principal de la aplicación debemos dirigarnos a la dirección:

<http://yob.ymca.org.es>

Allí lo primero que se nos mostrará será la ventana de que nos permitirá identificarnos en la aplicación.




Captura 1: Cuadro de Login

En cuanto se te conceda acceso se te entregará un nombre de usuario y contraseña que deberá introducir ahora.



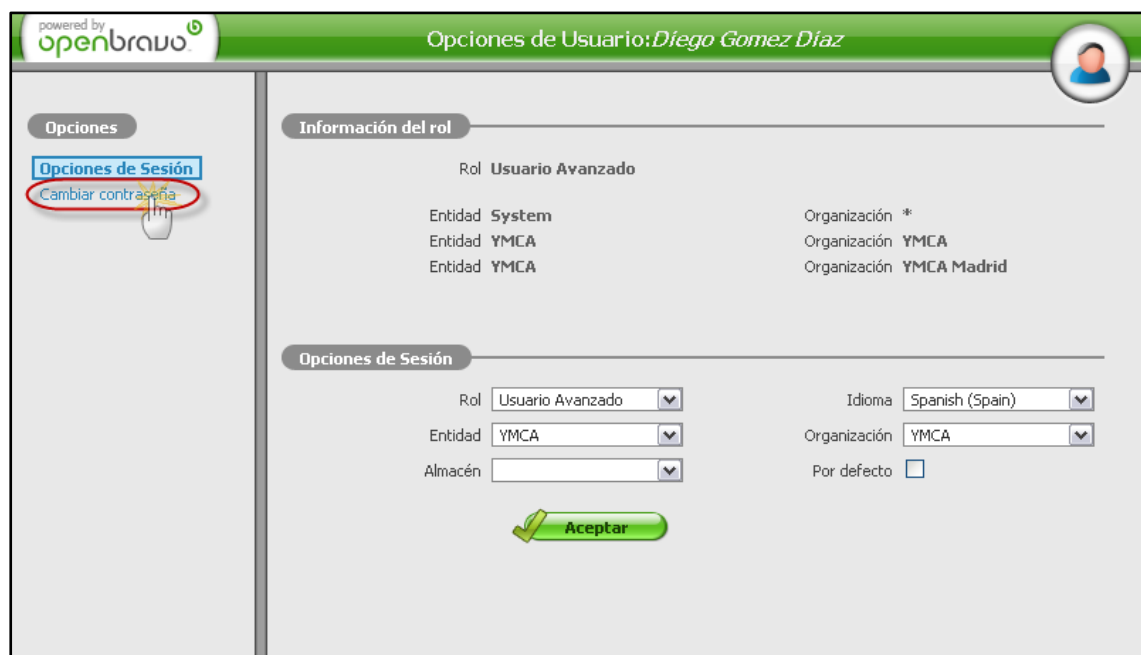
Captura 2: Cerrar sesión

3.4 Cambio de Contraseña

Haga clic en el botón de Preferencias de Usuario  y vaya a la opción Cambiar contraseña.

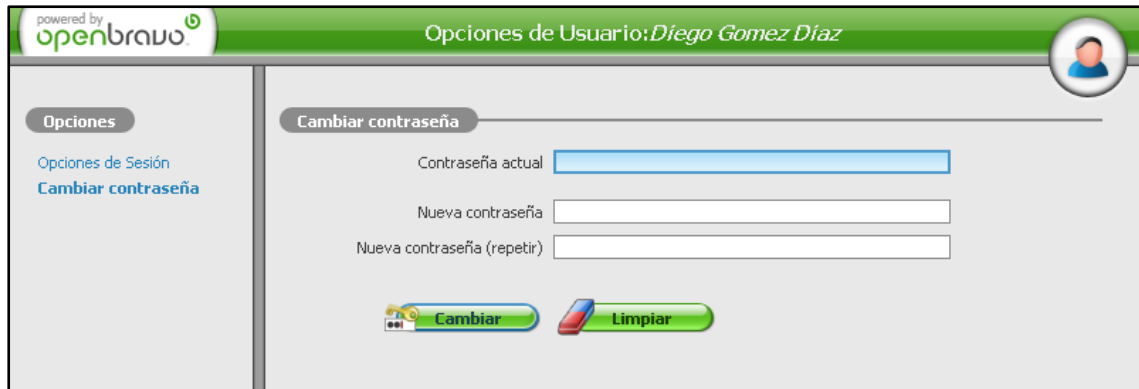


Captura 3: Opciones de la sesión



Captura 4: Cambio de contraseña

En la ventana se nos pedirá la contraseña actual y la nueva. Para evitar que otros usuarios nos cambien la contraseña en un momento de descuido.

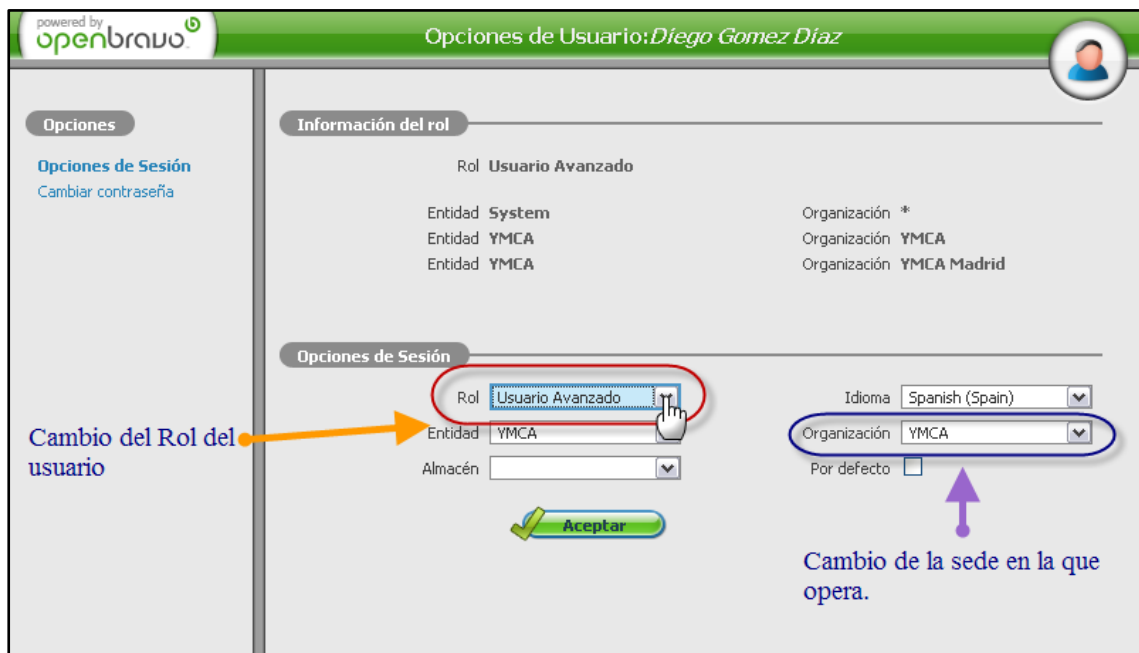


Captura 5: Introducir la nueva contraseña

Si pierde la contraseña debe solicitar una nueva al administrador del sistema o un usuario avanzado.

3.5 Cambio de Rol

El rol nos permite tener acceso a una u otra funcionalidad. Por ejemplo un rol de Responsable de Sede puede concedernos acceso a información financiera.



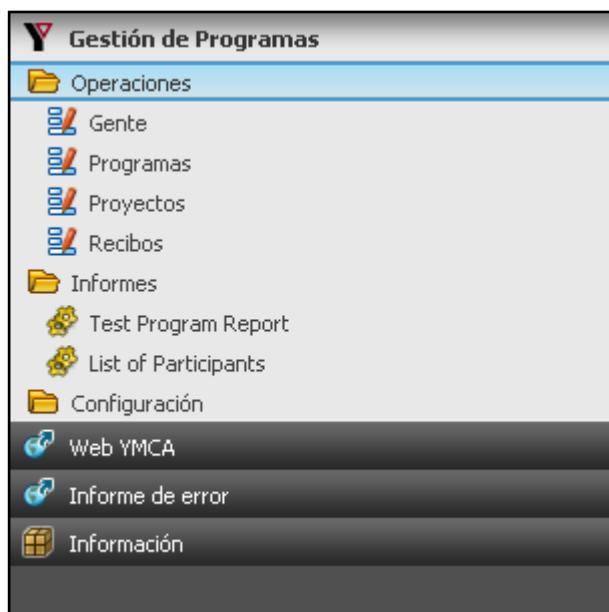
Captura 6: Cambio de rol

3.6 Interfaz

En esta sección se describen los elementos fundamentales de la interfaz de usuario de la aplicación YMCA Openbravo.







3.6.1 Menú de herramientas

El menú lateral nos muestra las herramientas que tenemos disponibles. Este listado variará en función de los permisos que tengamos asignados.



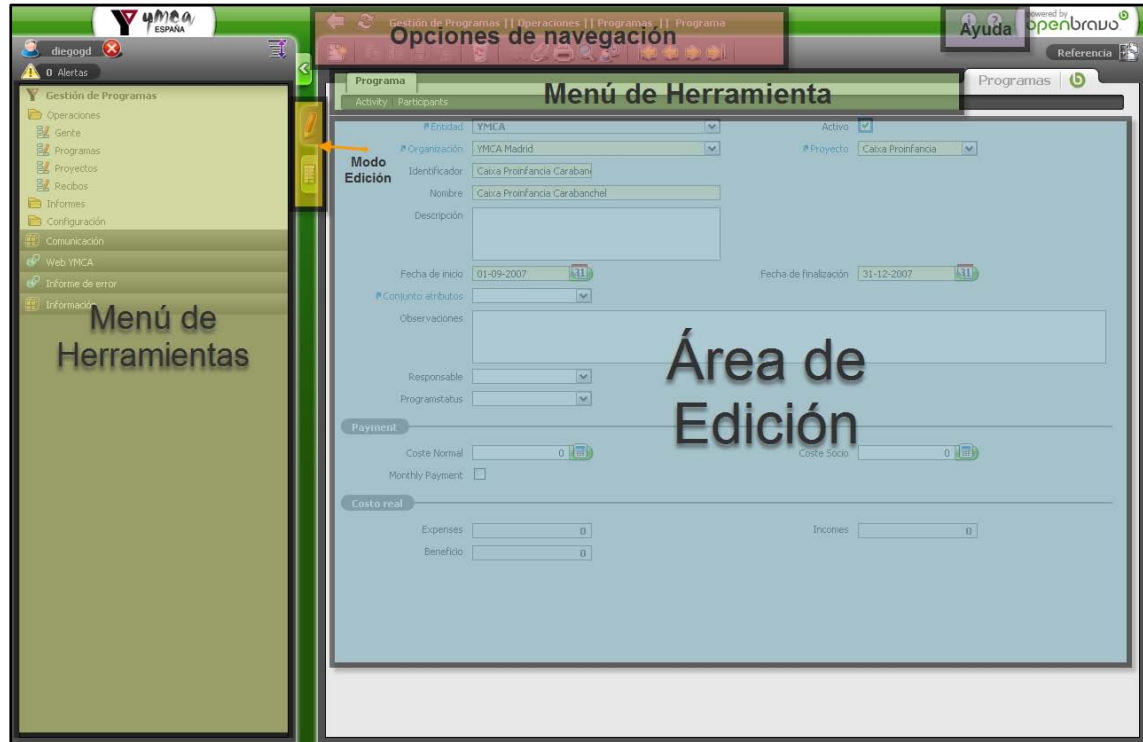
Captura 7: Menú de la aplicación

Los iconos de cada herramienta muestran la funcionalidad o frecuencia, como se muestra en la tabla siguiente.

	Carpetas
	Formulario
	Procesos o Informes
	Direcciones Web externas
	Flujos de trabajo
	Informe

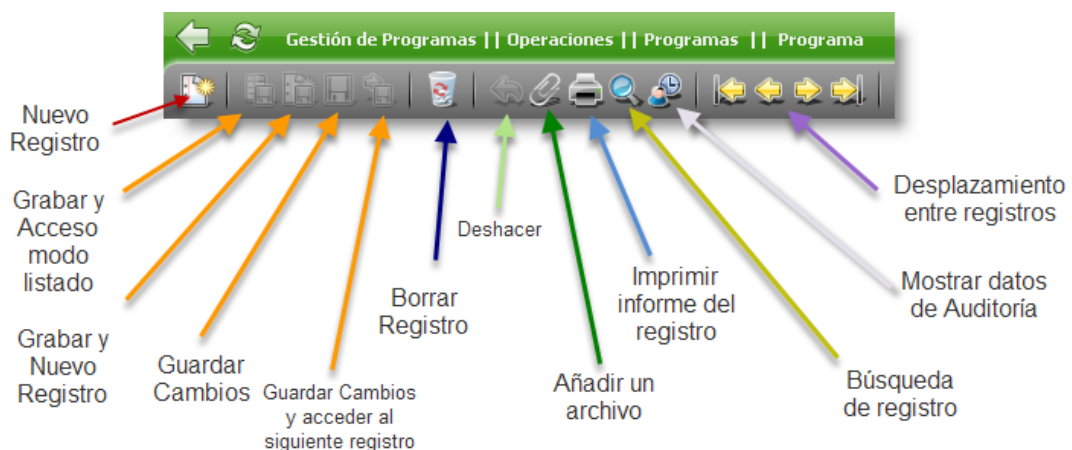
3.6.2 Controles de navegación

La aplicación está distribuida en diferentes áreas que se describen en la Captura 8.



Captura 8: Áreas de la ventana de la aplicación

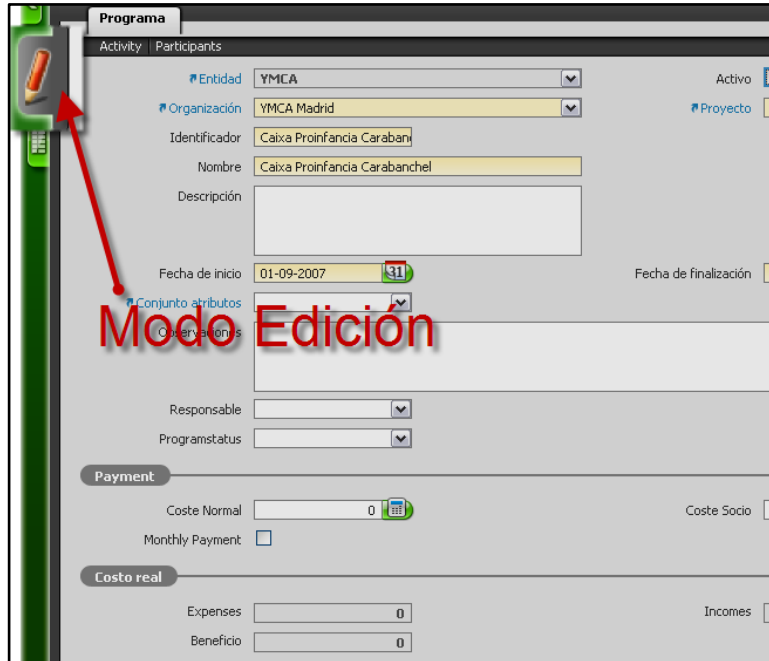
Dependiendo de la herramienta utilizada podremos visualizar todas o tan sólo algunos de los iconos descritos en la siguiente imagen.



Captura 9: Barra de tareas

El modo de edición / listado nos permite alternar entre dos formas de trabajo. Cuando necesitemos acceder al detalle de un solo registro o modificar su información

activaremos el modo Edición. Cuando realicemos una búsqueda o queramos generar un listado usaremos el modo listado (Parecido a Excel).



Programa

Activity | Participants

Entidad: YMCA

Organización: YMCA Madrid

Identificador: Caixa Proinfancia Carabanchel

Nombre: Caixa Proinfancia Carabanchel

Descripción:

Fecha de inicio: 01-09-2007

Fecha de finalización:

Responsable:

Programstatus:

Payment

Coste Normal: 0

Coste Socio:

Monthly Payment:

Costo real

Expenses: 0

Beneficio: 0

Incomes:

Captura 10: Activación del modo Edición

El modo listado nos permite además ordenar los registros por un determinado campo, hacer un filtrado a través de la función de búsqueda, y exportar el resultado a Excel o imprimirlo.



Programa

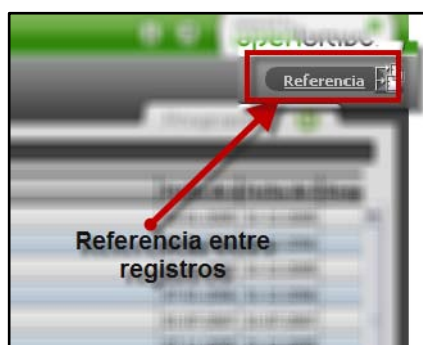
Activity | Participants

	Acta	Proyecto	Identificador	Nombre	Descripción
1	Y	Campamentos	1ª Agosto 05 Casa	1ª Agosto 05 Casavieja (Ávila)	
2	Y	Campamentos	1ª Agosto 06 Casa	1ª Agosto 06 Casavieja (Ávila)	
3	Y	Curso 2004/2005	1ª Julio 05 Casavieja	1ª Julio 05 Casavieja (Ávila)	
4	Y	Campamentos	1ª Julio 06, Casavieja	1ª Julio 06, Casavieja (Ávila)	
5	Y	Curso 2006/2007	1ª Quinc. Bembibre	1ª Quinc. Bembibre	
6	Y	Curso 2004/2005	2ª Decena Palencia	2ª Decena Palencia (Cuenca)	
7	Y	Campamentos	2ª Julio 06, Casavieja	2ª Julio 06, Casavieja (Ávila)	
8	Y	Curso 2004/2005	Arco Iris	Arco Iris	
9	Y	Curso 2004/2005	Arco Iris	Arco Iris	
10	Y	Curso 2006/2007	Bembibre 1ª quincena	Bembibre 1ª quincena Agosto	
11	Y	Infancia	Beniganim 2008 2ª quincena	Beniganim 2008 2ª quincena	
12	Y	Caixa Proinfancia	Caixa Proinfancia Carabanchel	Caixa Proinfancia Carabanchel	
13	Y	Infancia	Campa 2008 Ezcaray	Campa 2008 Ezcaray	
14	Y	Infancia	Campamento 08 Urbano Leganes	Campamento 08 Urbano Leganes	
15	Y	Curso 2005/2006	Campamento De Verano -Casavieja	Campamento De Verano -Casavieja	
16	Y	Curso 2005/2006	Campamento Semana Santa 2006	Campamento Semana Santa 2006	
17	Y	Campamentos	Campamento S. Santa 06	Campamento S. Santa 06	
18	Y	Curso 2005/2006	Campa Urbano Leganes	Campa Urbano Leganes 06	

Captura 11: Activación del modo Listado

3.6.3 Referencias entre registros

En YMCA Openbravo existen múltiples relaciones entre los diferentes registros, como un P participante y sus inscripciones. Este hecho hay que tenerlo en cuenta ya que la aplicación no dejará borrar registros. Además el conocer este atajo permite moverse mucho más rápido en determinados casos.



Captura 12: Visualizar Referencias entre registros

En las Captura 12 y Captura 13 vemos cómo se accede a esta función y cómo es visualizada. En el ejemplo vemos la relación que existe para el programa “Caixa Proinfancia Carabanchel” que tiene 85 participantes inscritos.



Captura 13: Venta de elementos relacionados

3.7 Estructura y nomenclatura de la aplicación

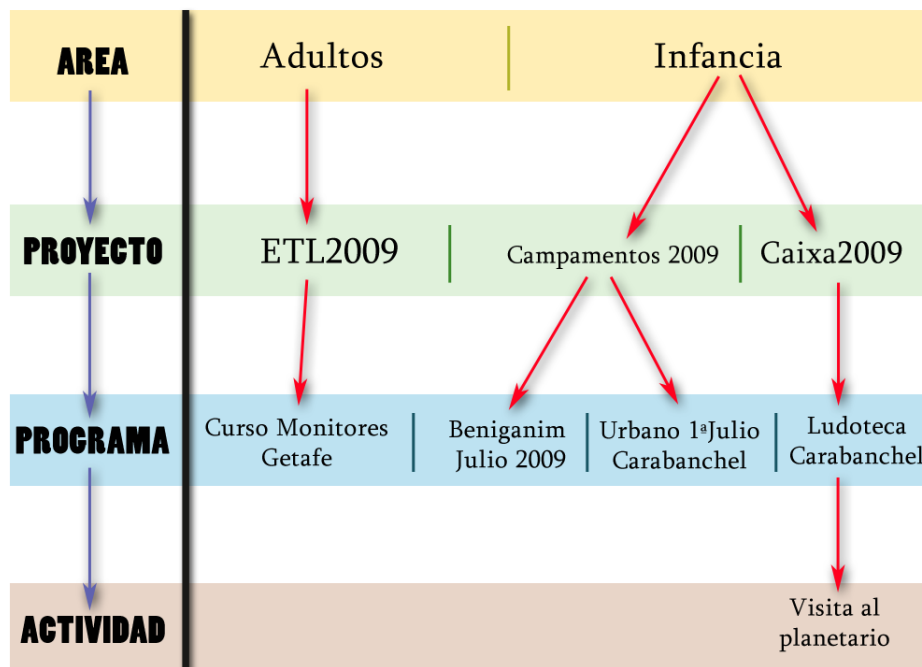


Ilustración 45: Dependencia entre elementos del sistema

3.8 Herramientas de gestión

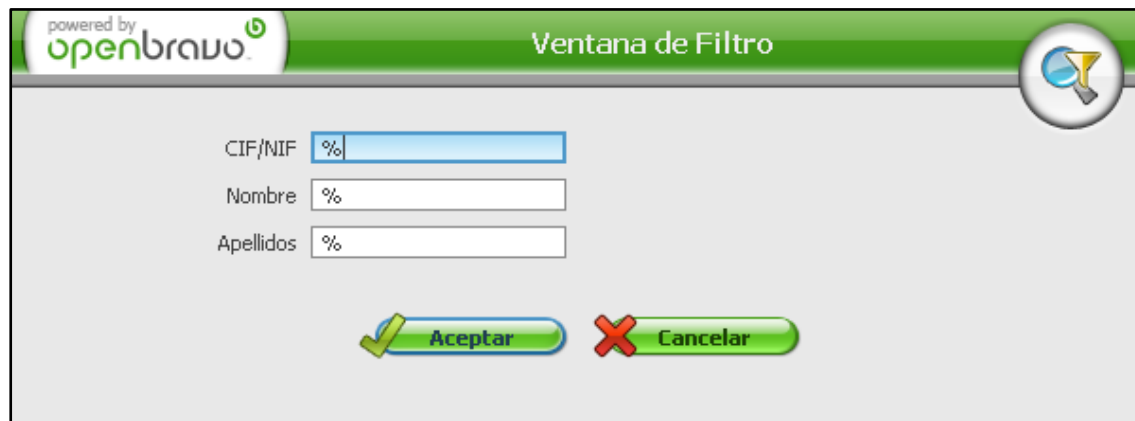
La funcionalidad principal que ofrece la aplicación es la siguiente:

- Alta y gestión de Proyectos agrupados en Áreas.
- Alta y gestión de Programas agrupados en Proyectos.
- Alta y gestión de Actividades agrupados en Proyectos.
- Alta y gestión de Participantes, Técnicos y Voluntarios.
- Inscripción de Participantes a programas y actividades.
- Generación de recibos de pago de programas y/o actividades.

Todos estos procesos se describen en detalle en el apartado 4 - Procesos habituales.

3.9 Herramienta de búsqueda

Dentro de los registros de la aplicación es posible realizar búsquedas por determinados campos. Al acceder a algunos registros, como participantes, se mostrará automáticamente el campo de búsqueda. Si no se puede acceder a él pulsando el icono:



Captura 14: Ventana de Búsqueda

Cuando se desconoce cierta información al realizar la búsqueda, es posible reemplazar cualquier campo por el comodín '%', igualmente este puede anteceder o suceder a cualquier palabra para indicar que coincidiría con cualquier cosa.

Por ejemplo: "ada%" localizaría "Ada Merced", pero "ada" sólo no.

3.10 Informes y listados

La herramienta además permite generar informes y listados de los diferentes elementos del sistema. Tales como:

- Listado de participantes por programa, actividad, edades o criterios de búsqueda.
- Listado de actividades, programas, proyectos y áreas por diferentes criterios.
- Listados de mailing.

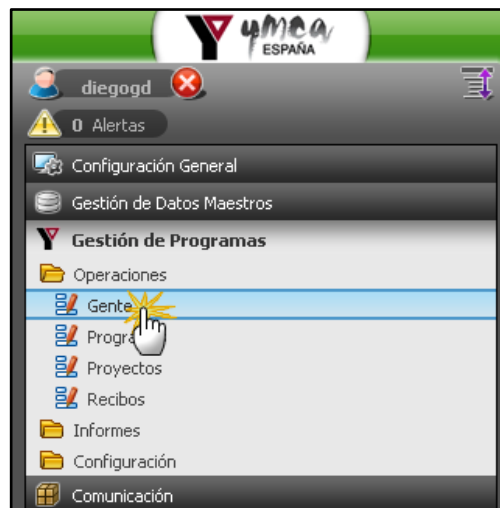
Sin embargo la herramienta en este sentido está a la espera de ser utilizada y ampliarse de acuerdo a las necesidades que vayan surgiendo, proceso que se describe en el apartado 6.2.

4 Procesos habituales

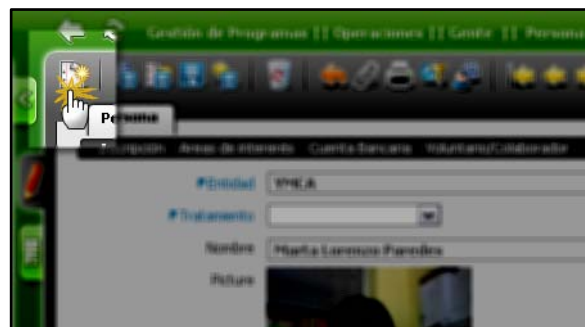
En este capítulo se repasan los procesos habituales con los que un trabajador podría encontrarse en su día a día en el uso del sistema.

4.1 Alta de participante

La gestión de los participantes es una de las claves de la aplicación, una vez tengamos un participante dado de alta en el sistema podremos inscribirle a programas y actividades.



Captura 15: Selección de la herramienta de edición de Participantes



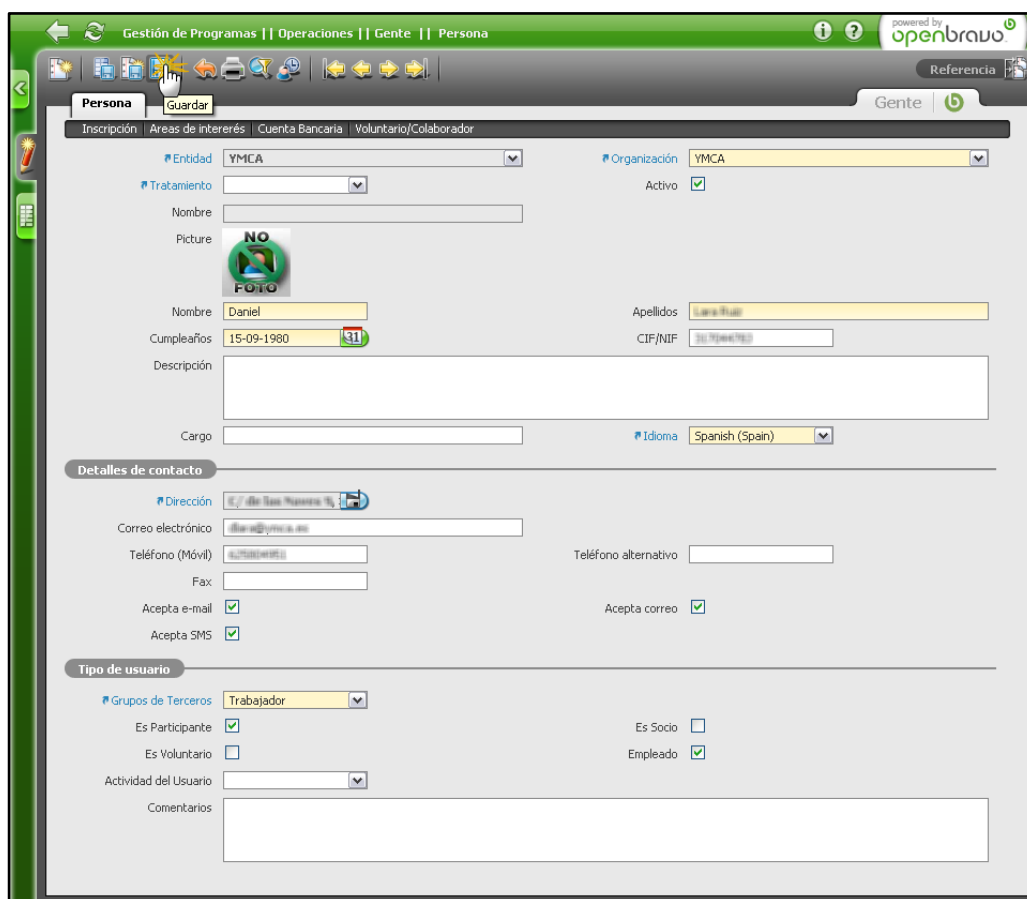
Captura 16: Crear un nuevo registro de Participante

Se deben rellenar de la ficha todos los campos que aparecen en amarillo y la dirección del participante.



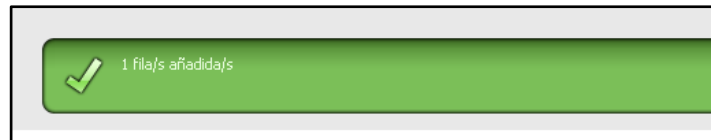
Captura 17: Ventana de edición de dirección postal

Una vez relleno todo es posible guardar el registro pulsando el icono con forma de floppy disk.



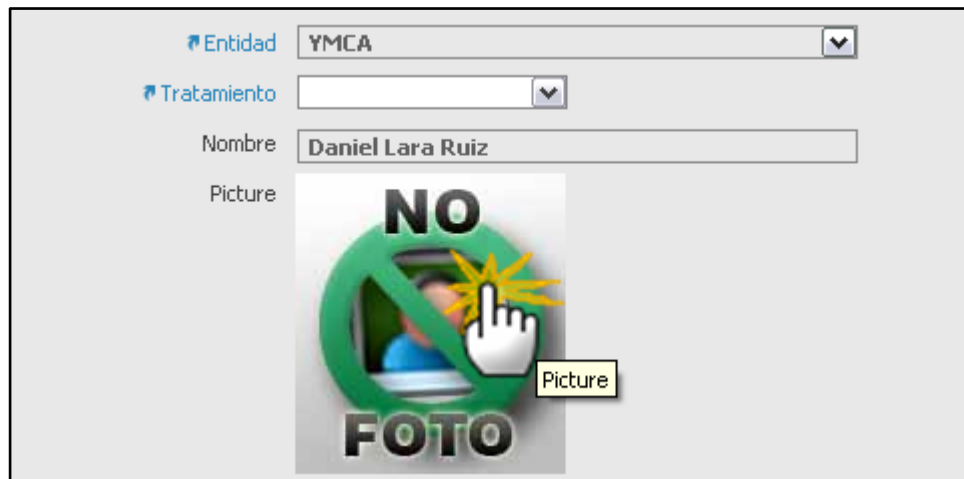
Captura 18: Tras rellenar la ficha guardamos los datos

El sistema notifica con un mensaje si el registro se ha insertado correctamente, o por el contrario ha habido algún fallo producido por la falta de algún campo o por conflicto con otro participante (por introducir dos veces un mismo registro).



Captura 19: Mensaje de notificación

Una vez guardado el participante es posible insertar o cambiar su fotografía pulsando en el icono que se muestra en la Captura 21.



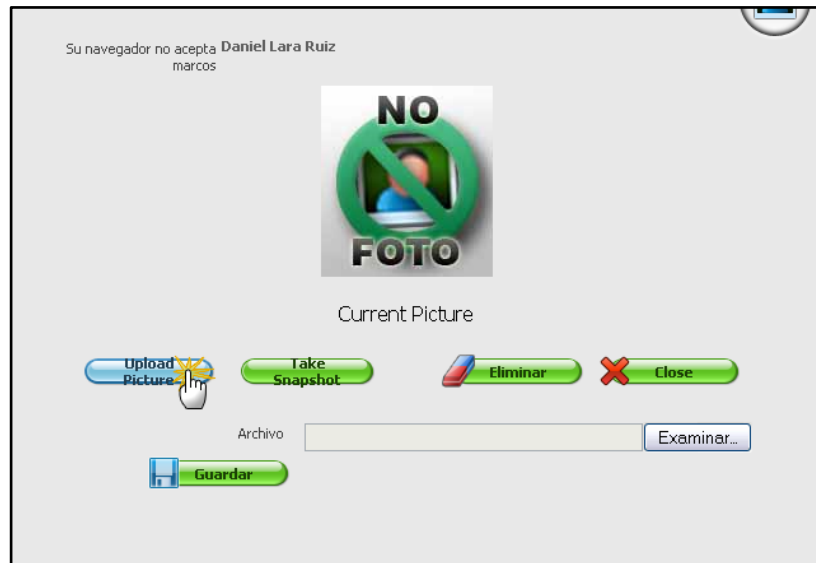
Captura 20: Pulsamos la fotografía para cambiar la imagen del Participante

Así se nos abrirá una ventana que nos permite introducir la fotografía con dos métodos alternativos.



Captura 21: Ventana de inserción de fotografía

El primer método permite seleccionar la fotografía a partir de un archivo. Este archivo será redimensionado a las dimensiones máximas permitidas automáticamente.



Captura 22: Cargando la imagen desde un archivo

El segundo método permite introducir la fotografía por medio de una webcam conectada al equipo. La interfaz utiliza un elemento Flash, por lo que es necesario tener instalado este componente en el equipo (Macromedia Flash).

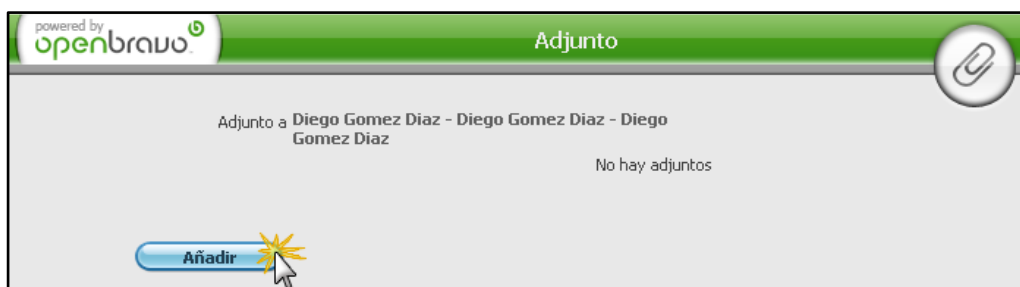


Captura 23: Cargando la imagen desde la webcam

Además es posible adjuntar archivos a las fichas del participante, para por ejemplo adjuntar su tarjeta de la seguridad social escaneada o el título de un monitor de tiempo libre.

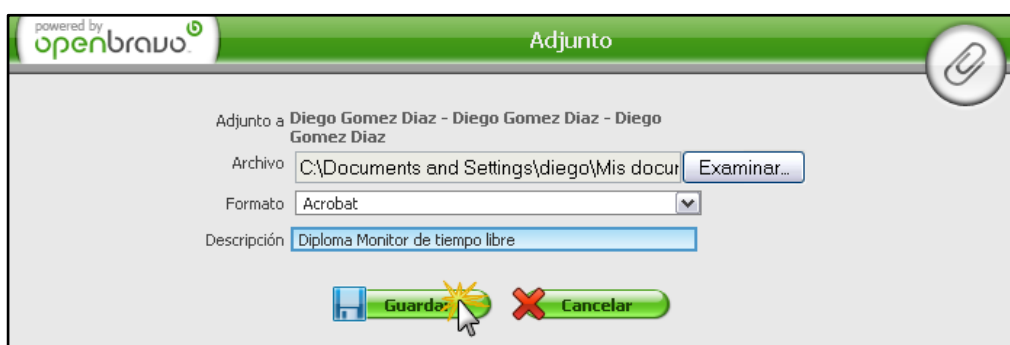


Captura 24: Adjuntar un archivo al registro



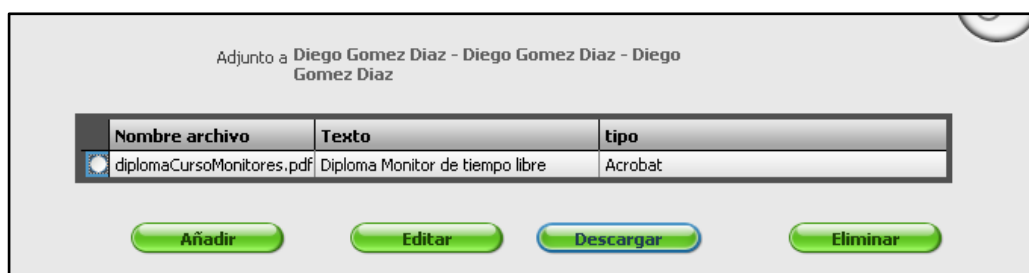
Captura 25: Ventana de adjuntos

Al añadir un archivo habrá que indicar la localización en el disco duro, el tipo de archivo y una descripción.



Captura 26: Selección del archivo a cargar

Una vez el registro contenga algún archivo, estos se mostrarán en una tabla como se muestra en la Captura 27.



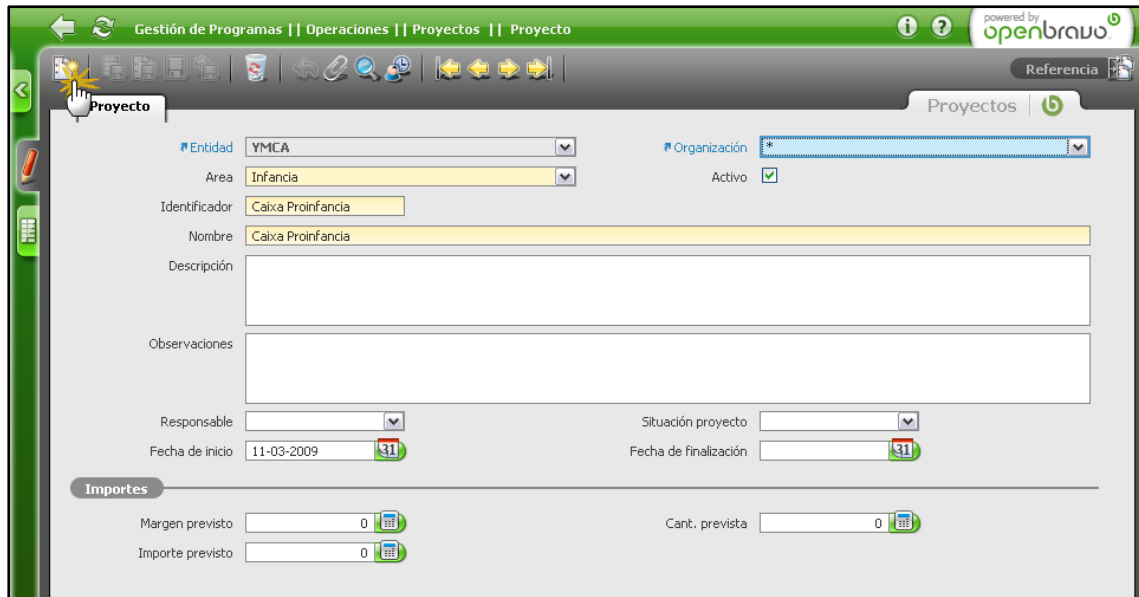
Captura 27: Listado de archivos adjuntos



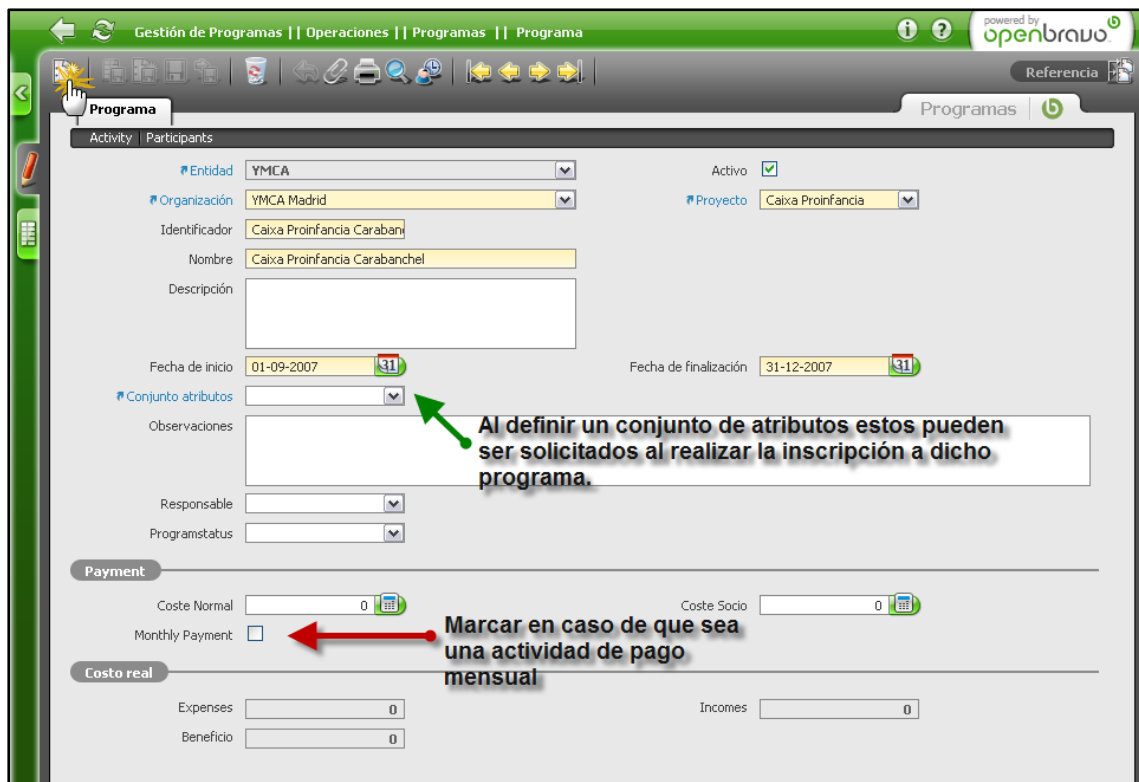
Se ha de tener cuidado con el tamaño de los archivos al añadirlos a un nuevo registro. Si se cargan numerosos archivos grandes se podría llegar a superar la capacidad de almacenamiento del servidor.

4.2 Alta de proyectos, programa y actividades

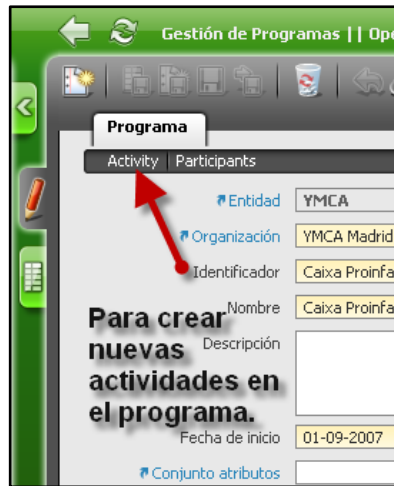
La creación de proyectos, programas y actividades es muy parecida, tan sólo hay que recordar cuál es la jerarquía que existe entre estos elementos (descrita en la sección 3.7).



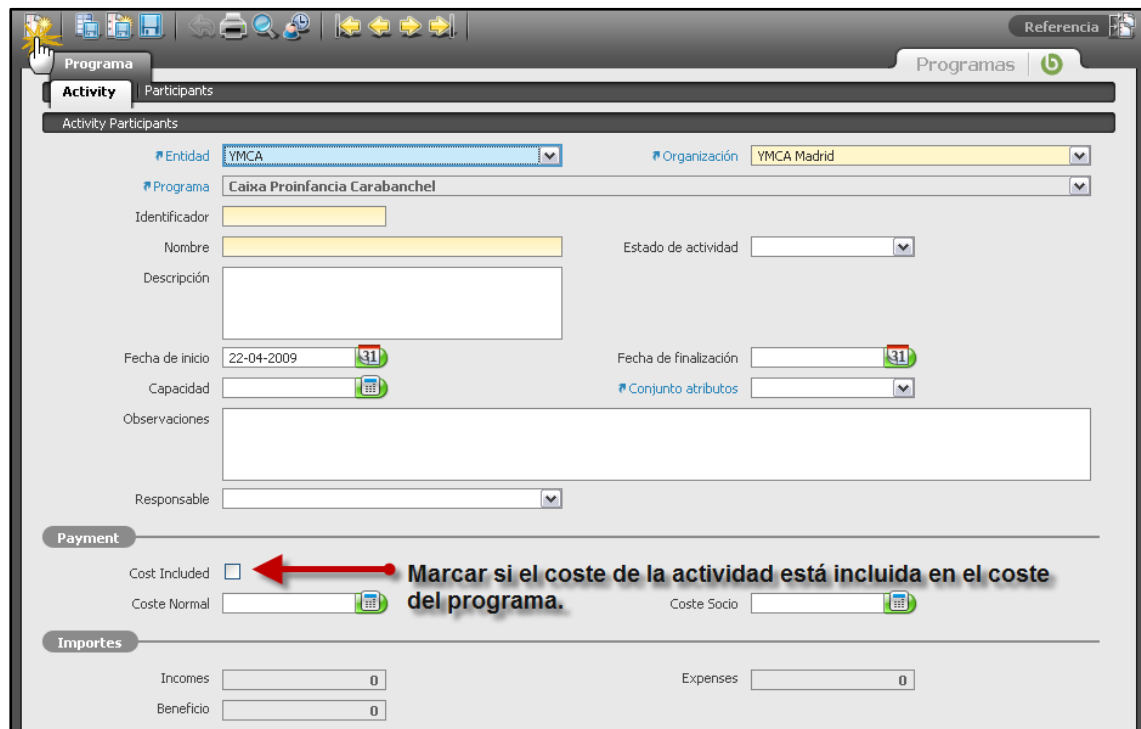
Captura 28: Alta de Proyecto



Captura 29: Alta de Programa



Captura 30: Selección de Actividades



Programa | **Activity** | Participants

Activity Participants

Entidad: YMCA | Organización: YMCA Madrid | Programa: Caixa Proinfancia Carabanchel

Identificador: | Nombre: | Descripción: | Estado de actividad: | Fecha de inicio: 22-04-2009 | Fecha de finalización: | Capacidad: | Conjunto atributos: | Observaciones: | Responsable: |

Payment

Cost Included: ☐ | Coste Normal: | Coste Socio: |

Importes

Incomes: 0 | Beneficio: 0 | Expenses: 0

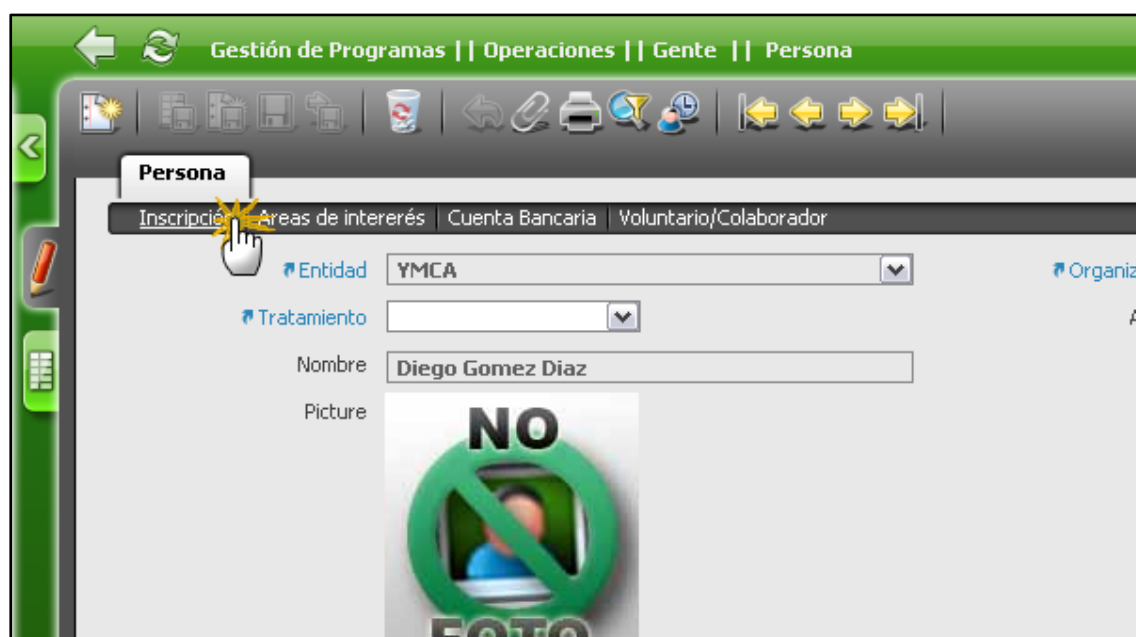
Marcar si el coste de la actividad está incluida en el coste del programa.

Captura 31: Alta de Actividad

4.3 Inscripción de participante

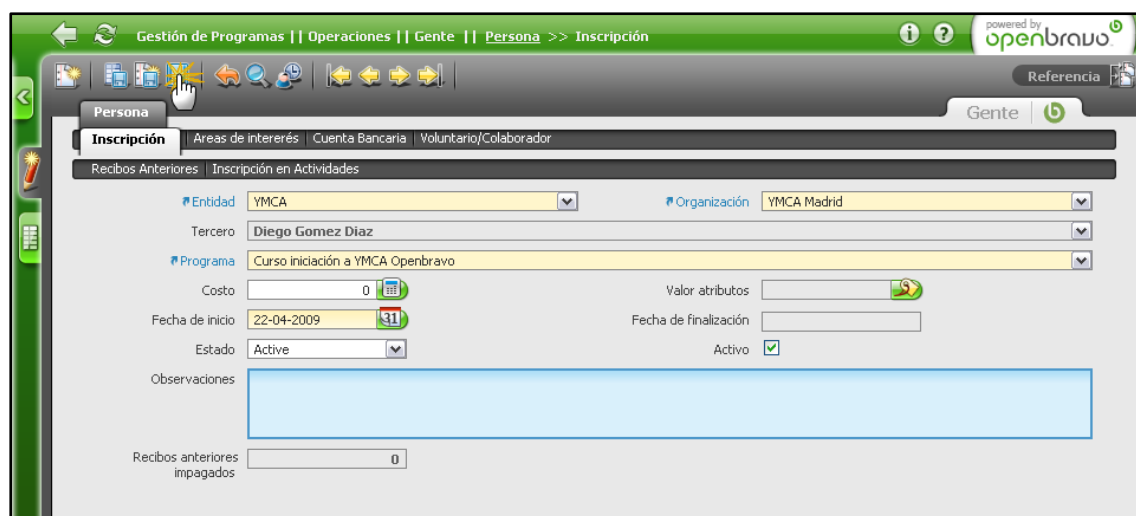
A través del alta de participantes en programas y actividades es posible llevar un seguimiento de la participación a través de listados y estadísticas.

Para inscribir un participante en una actividad, una vez que este haya sido dado de alta en el sistema, habrá que acceder a la opción Inscripción dentro de la ficha del Participante.



Captura 32: Inscripción de participante

Esa opción nos llevará al listado de todos los programas que tiene inscrito y además podemos añadir una nueva inscripción con el botón de “Nuevo Registro”.

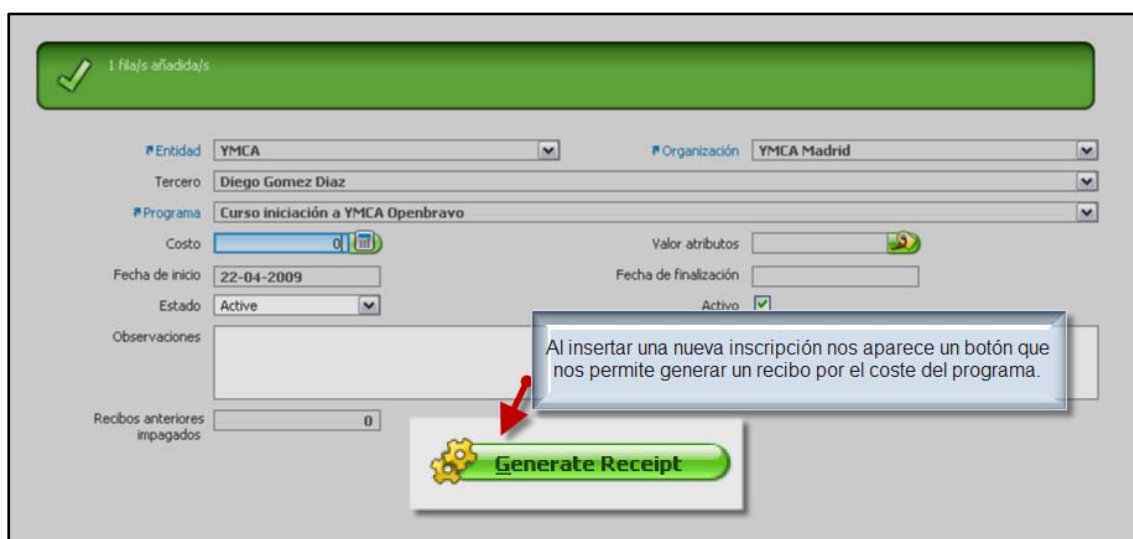


Captura 33: Guardando la inscripción

4.4 Generación de recibos

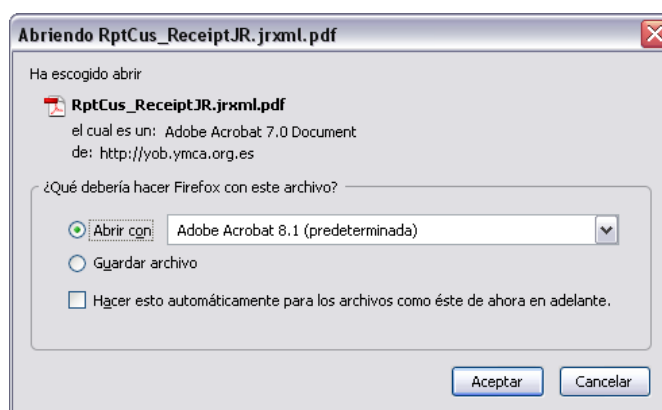
La generación de recibos es el paso habitual tras realizar una inscripción. A sí al grabar la inscripción nos aparecerá un botón que nos permite generar el recibo por la cantidad de Coste del Programa.

Es importante de que nos aseguremos que Organización tiene seleccionada nuestro usuario como predeterminada, ya que el recibo se creará con la información de esa Sede.

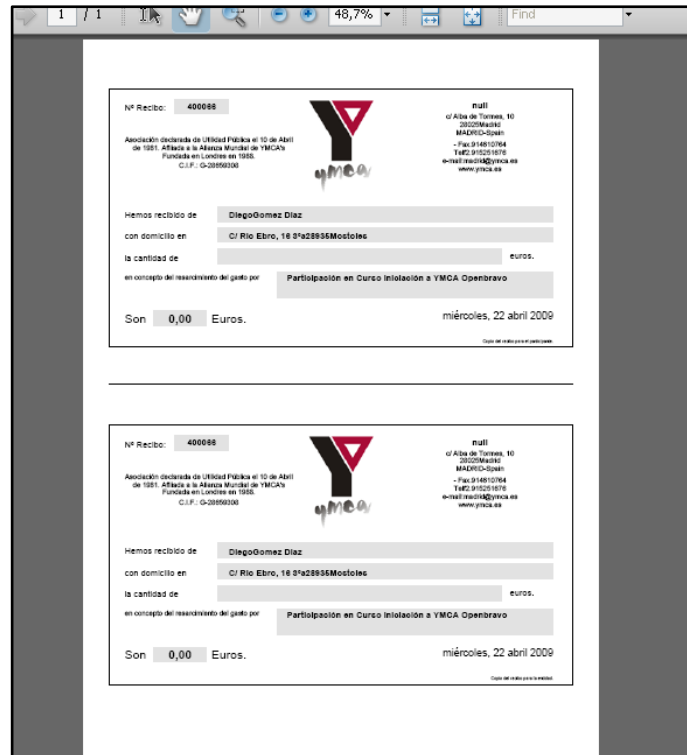


Captura 34: Generación de Recibo

El pulsar el botón el navegador descargará el PDF preguntándonos por la aplicación a utilizar.




Captura 35: Notificación de descarga de PDF



Captura 36: Es generado un archivo PDF con el recibo

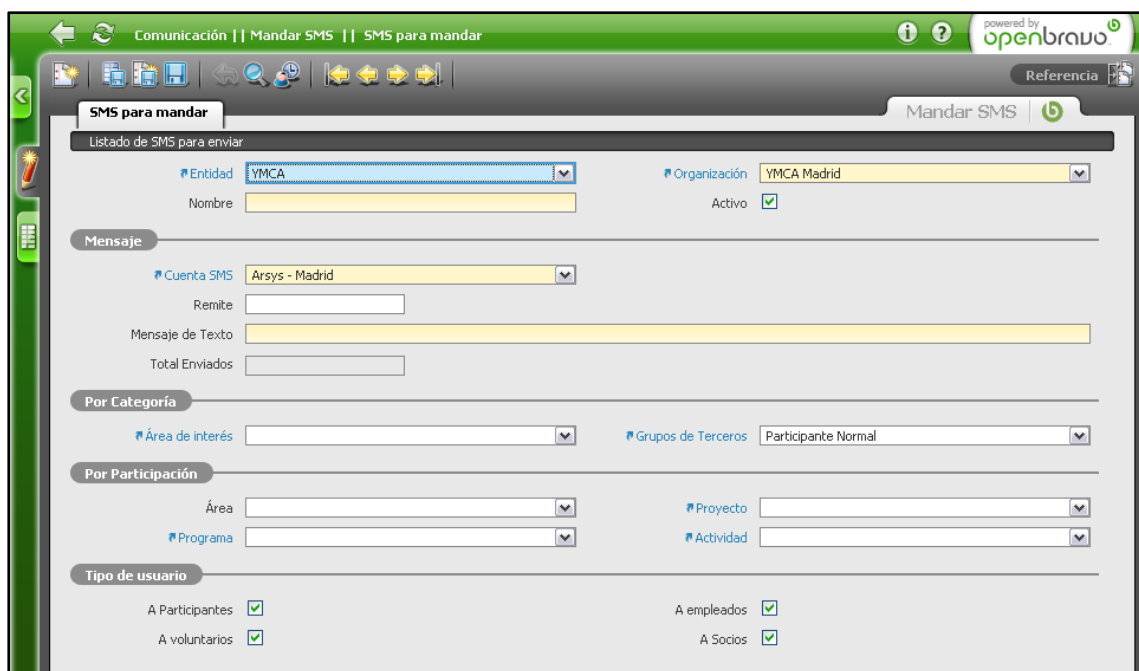
El aspecto del recibo es idéntico al que se producía con la anterior herramienta de YMCA Madrid (Captura 37).

Nº Recibo:	400066		YMCA Madrid c/ Alba de Tormes, 10 28025Madrid MADRID-Spain - Fax:914610764 Telf2.915251676 e-mail:madrid@ymca.es www.ymca.es
Asociación declarada de Utilidad Pública el 10 de Abril de 1981. Afiliada a la Alianza Mundial de YMCA's Fundada en Londres en 1988. C.I.F.: G-28659308			
Hemos recibido de	DiegoGomez Diaz		
con domicilio en	C/ Rio Ebro, 16 3ºa28935Mostoles		
la cantidad de			euros.
en concepto del resarcimiento del gasto por	Participación en Curso iniciación a YMCA Openbravo		
Son	0,00	Euros.	miércoles, 22 abril 2009
<small>Copia del recibo para el participante.</small>			

Captura 37: Apariencia del recibo

4.5 Envío de SMS a participantes

Una de las características más destacadas de la aplicación es el poder aprovechar los listados de participantes para generar envíos masivos de SMSs en unos pocos pasos que se describen a continuación.



Captura 38: Filtro de envío de SMS

Hay que tener en cuenta algunas consideraciones sobre el servicio de envío de SMS:

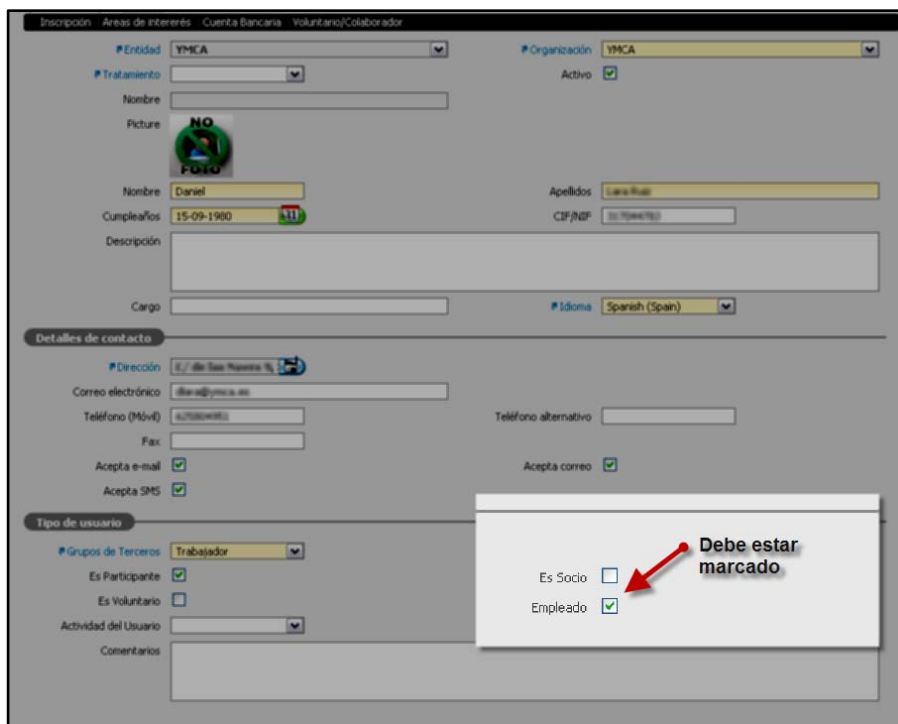
- Al tratarse de un servicio de pago es necesario hacer un uso responsable.
- Una vez enviado un SMS queda registrado con las notificaciones devueltas por el servidor (mensaje enviado o no enviado).
- El filtro de SMS es un medio para añadir varios usuarios al envío, sin embargo es posible añadir más manualmente antes de proceder al envío.
- En la configuración de la cuenta del servicio quedarán reflejados cuantos SMS quedan disponibles en el paquete contratado.

5 Herramientas administrativas

Los pasos que se describen a continuación son de interés para usuarios avanzados y administradores del sistema.

5.1 Alta de usuario

Ya vimos en la sección 4.1 - “Alta de participante” cómo dar de alta un participante en el sistema. En esta sección además veremos cómo un participante es la base para crear un usuario en el sistema.



The screenshot shows a web form for modifying a participant. The form includes fields for personal information (Name, Surname, Date of Birth, ID Number), contact details (Address, Email, Phone), and user type. In the 'Tipo de usuario' section, the 'Es Empleado' checkbox is checked, and a red arrow points to it with the text 'Debe estar marcado'.

Captura 39: Modificar Participante para añadir el Usuario

Una vez el usuario tenga sus datos declarados en su ficha de Participante podremos acceder al menú Configuración General/Seguridad/Usuario.



Captura 40: Accede a la edición de usuarios

Esta lista tan sólo los Trabajadores de la entidad, que serán los potenciales usuarios.

Configuración General || Seguridad || Usuario || Usuario

1 - 13 / 13

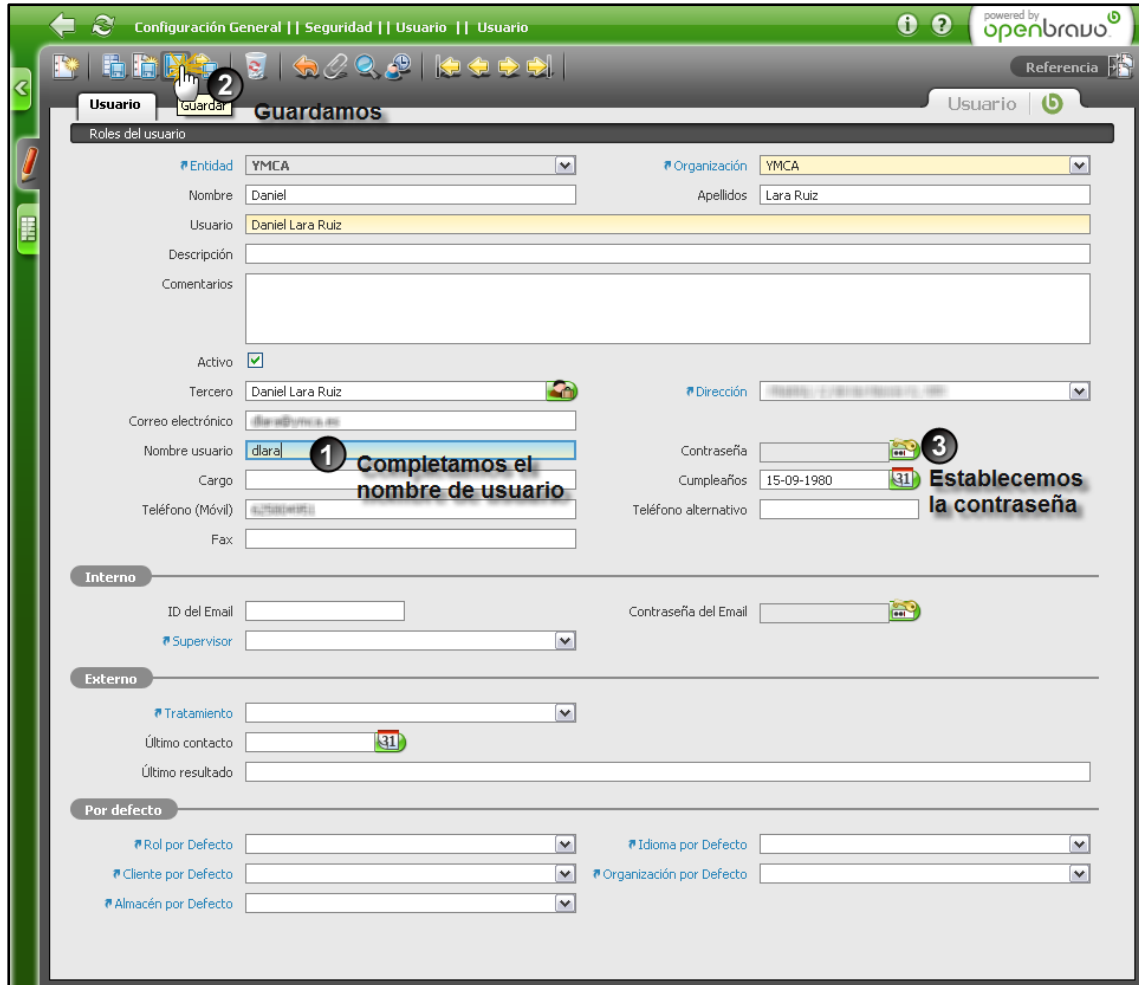
Usuario

Roles del usuario

	Nombre	Apellidos	Usuario
1	Ada Merced	Sanchez Feijoo	Ada Merced Sanchez Feijoo
2	Borja	Garcia Amador	Borja Garcia Amador
3	Daniel	Lara Ruiz	Daniel Lara Ruiz
4	Davide	Bombassei	Davide Bombassei -
5	Diego	Gomez Diaz	Diego Gomez Diaz
6	Estrella	Espada González	Estrella Espada González
7	Marta	Lorenzo Paredes	Marta Lorenzo Paredes
8	Openbravo		Openbravo
9	Tamara	Ortega Garcia	Tamara Ortega Garcia
10	usuario		usuario
11			System
12			YMCAAdmin
13			YMCAUser

Captura 41: Se listan los Participantes que son Trabajadores (casilla marcada)

Accederemos al usuario deseado para completar sus credenciales de acceso. Hasta entonces no podrá acceder al sistema.



Configuración General || Seguridad || Usuario || Usuario

powered by openbravo

Referencia

Usuario Guardar Usuario

Roles del usuario

Entidad YMCA Organización YMCA

Nombre Daniel Apellidos Lara Ruiz

Usuario Daniel Lara Ruiz

Descripción

Comentarios

Activo ☒

Tercero Daniel Lara Ruiz

Correo electrónico dlarar@ymca.es

Nombre usuario dlarar 1 Completamos el nombre de usuario

Cargo

Teléfono (Móvil)

Fax

Dirección

Contraseña 3 Establecemos la contraseña

Cumpleaños 15-09-1980

Teléfono alternativo

Interno

ID del Email

Contraseña del Email

Supervisor

Externo

Tratamiento

Último contacto 31

Último resultado

Por defecto

Rol por Defecto

Idioma por Defecto

Organización por Defecto

Cliente por Defecto

Almacén por Defecto

Captura 42: Insercción de "Nombre de Usuario" y "Contraseña"

Las credenciales se completarán con el nombre de usuario y contraseña. Tan solo hay que tener cuidado de no repetir el nombre de usuario (aunque la aplicación lo notificará).



powered by openbravo

Botones encriptados

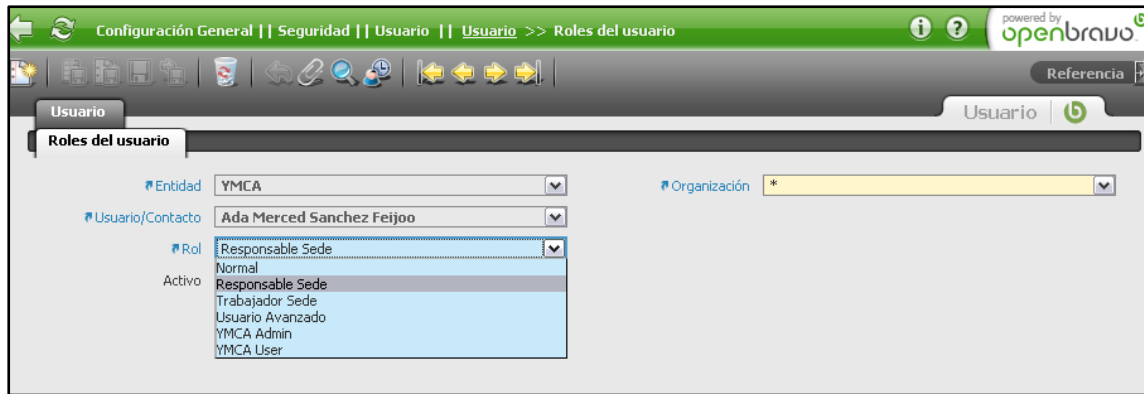
Contraseña

Aceptar Cancelar

Captura 43: Ventana de inserción de contraseña

Así creamos las credenciales del usuario para que pueda acceder al sistema. Sin embargo aún no podría acceder puesto que no tiene ningún rol asignado.

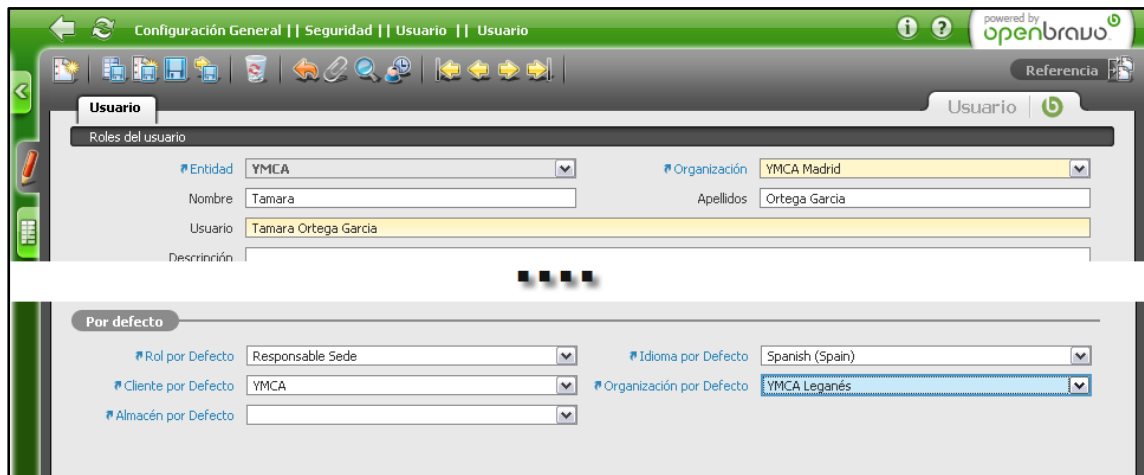
Para añadir un rol al usuario habrá que pinchar en la opción Roles del Usuario y asignar uno nuevo.



The screenshot shows the 'Roles del usuario' (User Roles) configuration page in Openbravo. The breadcrumb trail is 'Configuración General >> Seguridad >> Usuario >> Usuario >> Roles del usuario'. The page has a green header with the Openbravo logo. The main content area is divided into two sections: 'Entidad' (Entity) and 'Organización' (Organization). Under 'Entidad', there is a dropdown menu for 'Entidad' set to 'YMCA', a dropdown for 'Usuario/Contacto' set to 'Ada Merced Sanchez Feijoo', and a dropdown for 'Rol' with a list of roles: 'Responsable Sede', 'Normal', 'Responsable Sede', 'Trabajador Sede', 'Usuario Avanzado', 'YMCA Admin', and 'YMCA User'. The 'Organización' section has a dropdown for 'Organización' set to '*'. There is also a 'Referencia' (Reference) button in the top right corner.

Captura 44: Asignación de Rol a Usuario

Posteriormente es posible definir el rol y sede por defecto para el usuario, volviendo a la configuración del Usuario.

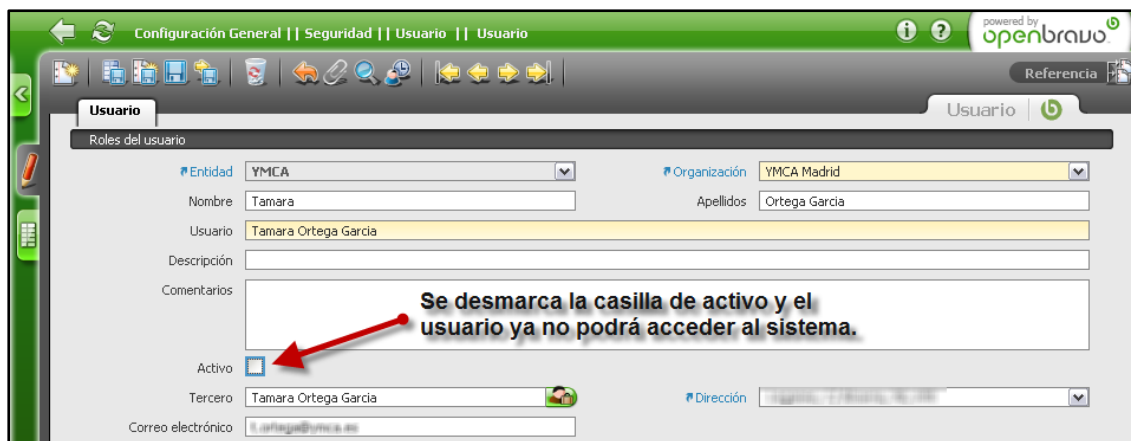


The screenshot shows the 'Por defecto' (Default) configuration page in Openbravo. The breadcrumb trail is 'Configuración General >> Seguridad >> Usuario >> Usuario'. The page has a green header with the Openbravo logo. The main content area is divided into two sections: 'Roles del usuario' (User Roles) and 'Por defecto' (Default). Under 'Roles del usuario', there is a dropdown for 'Entidad' set to 'YMCA', a dropdown for 'Organización' set to 'YMCA Madrid', and text fields for 'Nombre' (Name) set to 'Tamara', 'Apellidos' (Surnames) set to 'Ortega Garcia', and 'Usuario' (Username) set to 'Tamara Ortega Garcia'. There is also a 'Descripción' (Description) field. Under 'Por defecto', there are dropdowns for 'Rol por Defecto' (Default Role) set to 'Responsable Sede', 'Idioma por Defecto' (Default Language) set to 'Spanish (Spain)', 'Cliente por Defecto' (Default Client) set to 'YMCA', and 'Organización por Defecto' (Default Organization) set to 'YMCA Leganés'. There is also a dropdown for 'Almacén por Defecto' (Default Warehouse) set to an empty value. There is a 'Referencia' (Reference) button in the top right corner.

Captura 45: Opciones por defecto

5.2 Baja de usuario

En el sistema una vez es creado un usuario no es conveniente su eliminación puesto que existen multitud de relaciones que han sido creadas durante la vida del usuario en el sistema, por tanto en lugar de borrarlo del sistema simplemente se desactivará.



Captura 46: Desactivar usuario del sistema

5.3 Gestión de roles

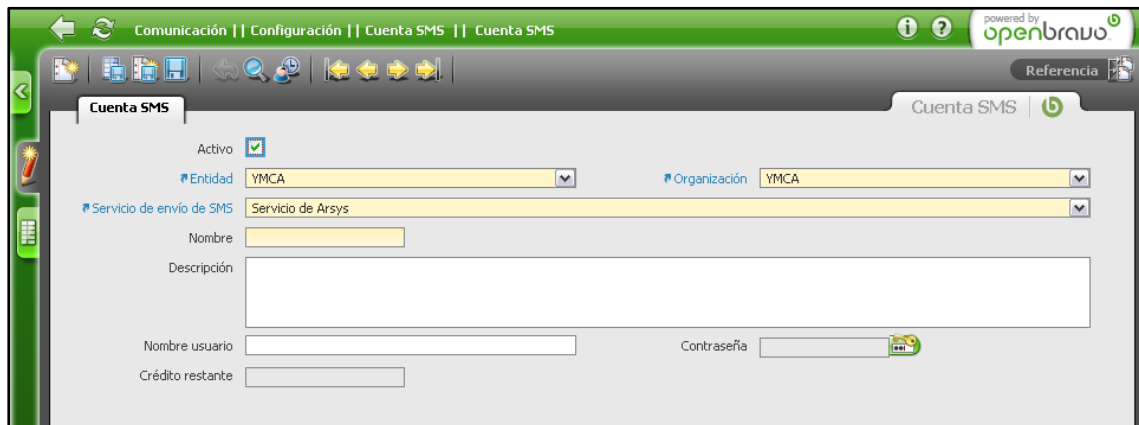
El rol de un usuario determina sus permisos de acceso a la aplicación. Un usuario puede disponer de varios roles según la necesidad, e ir cambiando de uno a otro de acuerdo a las actividades que vaya a realizar.

El rol define que ventanas y procesos tiene disponible un usuario y a qué organizaciones puede acceder.

En potencia habrá que declarar tantos roles como limitaciones quiera que exista en el sistema.

5.4 Configuración del servicio de SMS

El servicio de envío de SMS permite el uso de diferentes proveedores de este tipo de servicios, sin embargo actualmente sólo hay implementado uno de ellos. Para usar otro es necesaria una pequeña implementación que deberá ser solicitada por los métodos establecidos en la sección 6.2.



Captura 47: Alta de cuenta de Envío de SMS

6 Reporte de Errores

6.1 Necesidad

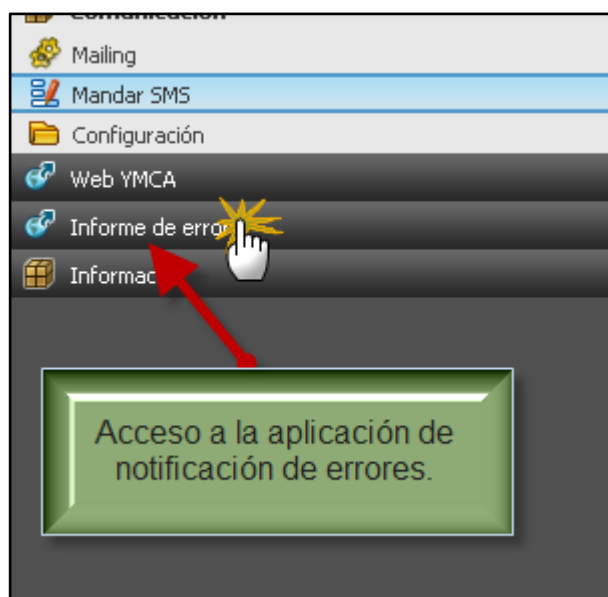
La aplicación está en una etapa temprana de desarrollo, aunque es totalmente funcional existen errores que aún no han sido detectados.

Al poner la aplicación en producción es habitual que estos errores aparezcan motivo del uso poco predecible de cada tipo de usuario. Para mejorar la aplicación es necesario que estos errores sean notificados y para ello se ha instalado una aplicación que cubrirá esta función.

Además de los errores, como ya se describió antes, existen necesidades no cubiertas, fundamentalmente con los informes. Esta plataforma sirve igualmente para solicitar nuevas funcionalidades.

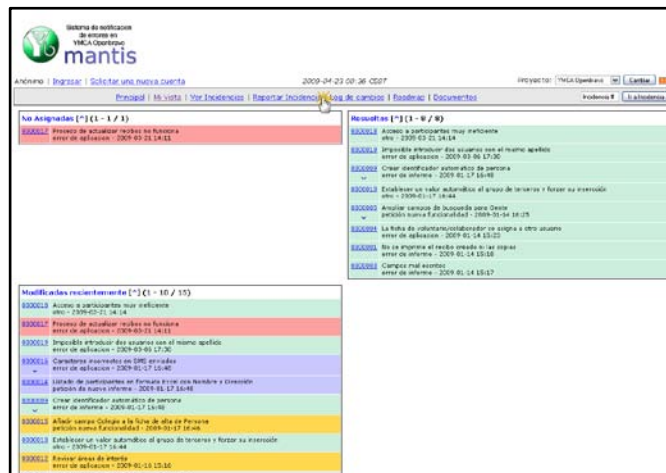
6.2 Procedimiento

Como el resto de las funciones, la aplicación de reportes de errores es accesible a través del menú de la aplicación.



Captura 48: Acceso a Mantis

Accederemos a una nueva página web que nos muestra el estado actual de errores abiertos y solucionados y las peticiones de nuevas funcionalidades.

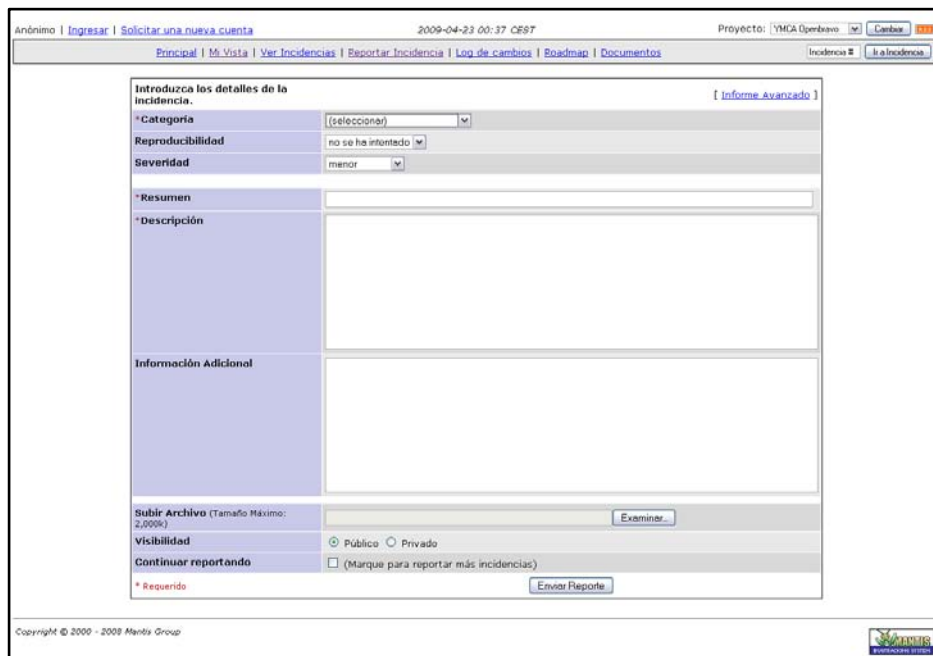


Ver Asignadas [1 (1 - 1 / 1)]		Resueltas [1 (1 - 6 / 6)]	
1000013	Acceso a participantes muy ineficiente error de aplicación - 2009-03-23 14:03	1000014	Acceso a participantes muy ineficiente voto - 2009-03-23 14:14
1000012	Proceso de activación no funciona error de aplicación - 2009-03-23 14:03	1000015	Imposible eliminar dos usuarios con el mismo apellido error de aplicación - 2009-03-26 17:30
1000011	Imposible eliminar dos usuarios con el mismo apellido error de aplicación - 2009-03-26 17:30	1000016	Crear identificador automático de persona error de aplicación - 2009-03-27 15:46
1000010	Establecer un valor automático al grupo de usuarios y hacer su inserción voto - 2009-03-27 15:46	1000017	Anular campos de acuerdo para crear proyecto nueva funcionalidad - 2009-03-28 10:25
1000009	Establecer un valor automático al grupo de usuarios y hacer su inserción error de aplicación - 2009-03-28 10:25	1000018	La fecha de voluntariado/colaborador se suma a otro usuario error de aplicación - 2009-03-28 10:25
1000008	Imposible eliminar dos usuarios con el mismo apellido error de aplicación - 2009-03-28 10:25	1000019	Imposible eliminar dos usuarios con el mismo apellido error de aplicación - 2009-03-28 10:25
1000007	Imposible eliminar dos usuarios con el mismo apellido error de aplicación - 2009-03-28 10:25	1000020	Imposible eliminar dos usuarios con el mismo apellido error de aplicación - 2009-03-28 10:25
1000006	Imposible eliminar dos usuarios con el mismo apellido error de aplicación - 2009-03-28 10:25	1000021	Imposible eliminar dos usuarios con el mismo apellido error de aplicación - 2009-03-28 10:25
1000005	Imposible eliminar dos usuarios con el mismo apellido error de aplicación - 2009-03-28 10:25	1000022	Imposible eliminar dos usuarios con el mismo apellido error de aplicación - 2009-03-28 10:25
1000004	Imposible eliminar dos usuarios con el mismo apellido error de aplicación - 2009-03-28 10:25	1000023	Imposible eliminar dos usuarios con el mismo apellido error de aplicación - 2009-03-28 10:25
1000003	Imposible eliminar dos usuarios con el mismo apellido error de aplicación - 2009-03-28 10:25	1000024	Imposible eliminar dos usuarios con el mismo apellido error de aplicación - 2009-03-28 10:25
1000002	Imposible eliminar dos usuarios con el mismo apellido error de aplicación - 2009-03-28 10:25	1000025	Imposible eliminar dos usuarios con el mismo apellido error de aplicación - 2009-03-28 10:25
1000001	Imposible eliminar dos usuarios con el mismo apellido error de aplicación - 2009-03-28 10:25	1000026	Imposible eliminar dos usuarios con el mismo apellido error de aplicación - 2009-03-28 10:25

Captura 49: Estado de peticiones

Si presionamos el link “Reportar Incidencia” accederemos a un formulario donde se nos pide que describamos el error o petición.

Es muy importante ser detallistas al rellenar el cuestionario ya que eso ayudará a su pronta resolución.



Introduzca los detalles de la incidencia. [Informe Avanzado]

*Categoría: (seleccionar)

Reproducibilidad: no se ha intentado

Severidad: menor

*Resumen:

*Descripción:

Información Adicional:

Subir Archivo (Tamaño Máximo: 2,000K) Examinar

Visibilidad: ☒ Público ☐ Privado

Continuar reportando: ☐ (Marque para reportar más incidencias)


* Requerido Enviar Reporte

Copyright © 2000 - 2008 Mantis Group

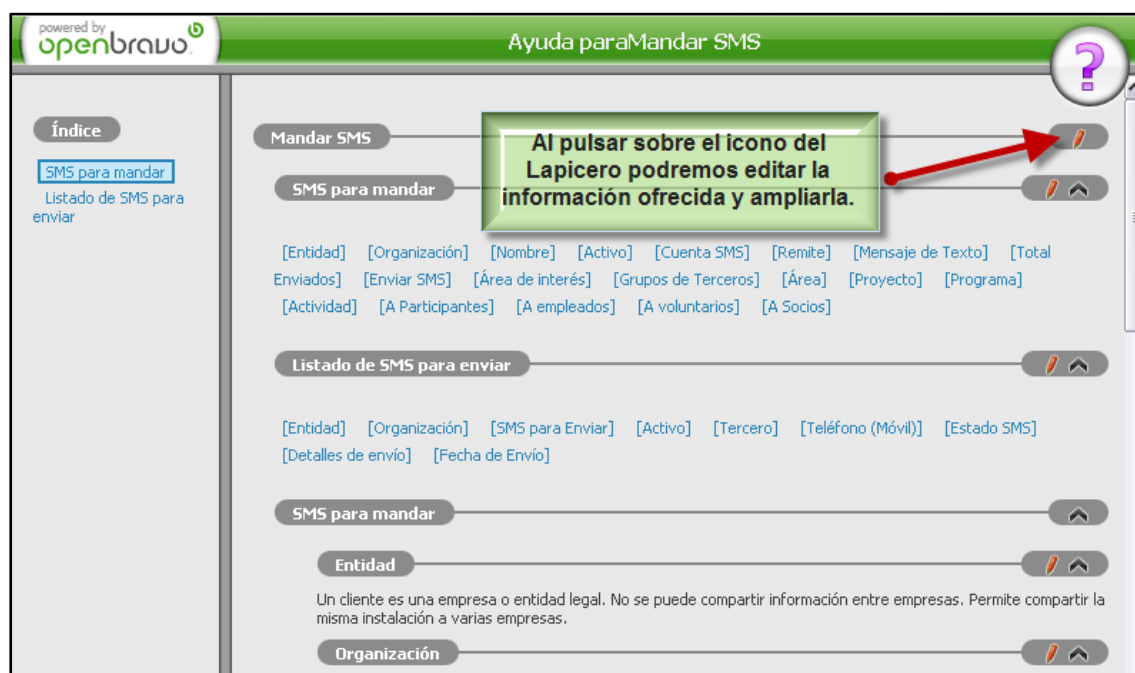
Captura 50: Informe de Notificación de errores o peticiones

7 Ampliando el conocimiento

7.1 Ayuda de la aplicación

En todo momento es posible obtener ayuda sobre las diferentes herramientas accediendo a la ayuda que se encuentra en esquina superior derecha: 

Esta ayuda puede ser ampliada por cualquier usuario a medida que es usada la aplicación.



Captura 51: Edición de ayuda

7.2 Curso auto-formativo

Se está desarrollando un curso online que repasará todos los temas contenidos en este manual, a través de videos e información ampliada.

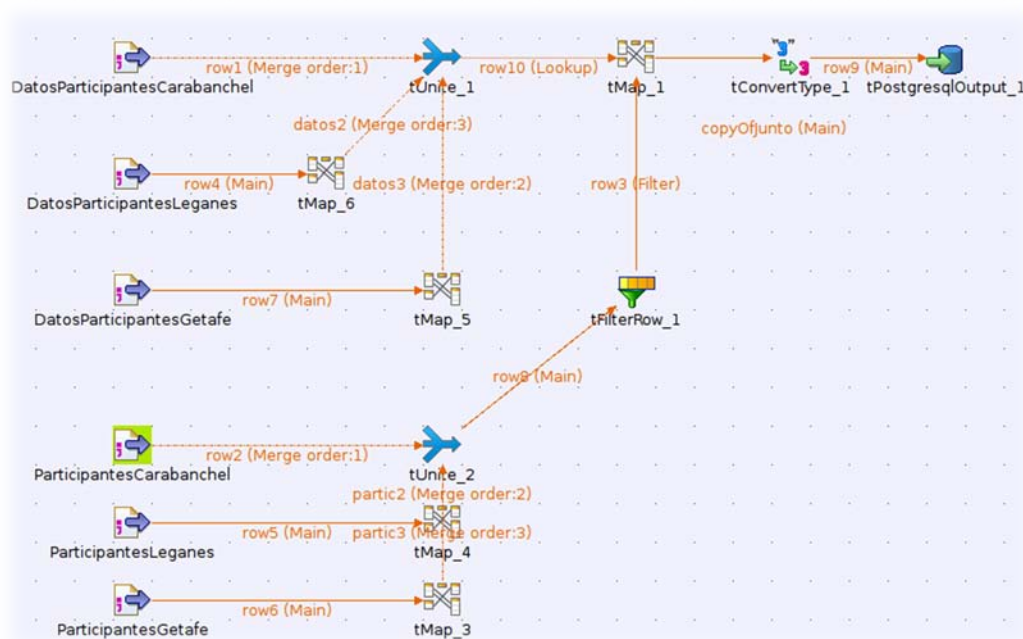
Este curso estará disponible a través de la plataforma online de formación Moodle de YMCA.

Anexo II – Proceso detallado de migración

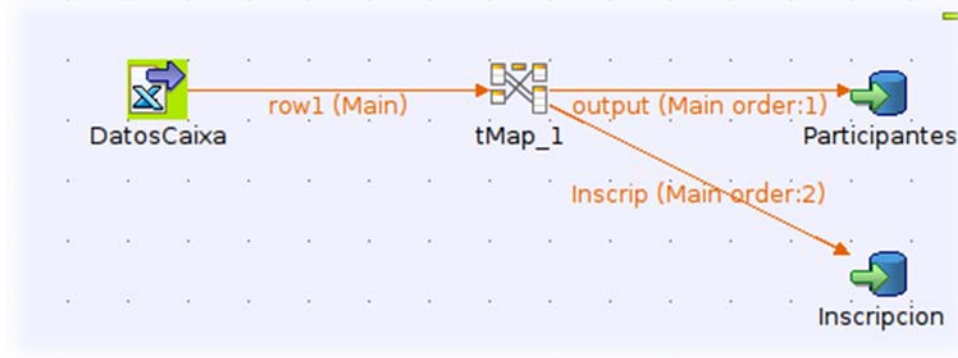
1 Migración al sistema productivo

A continuación se describe el proceso completo de migración de los datos del sistema antiguo de Y MCA a la aplicación. Es un proceso muy manual que conviene conservar para futuras consultas y su reproductividad.

Importar participantes desde las tres fuentes de datos existentes.



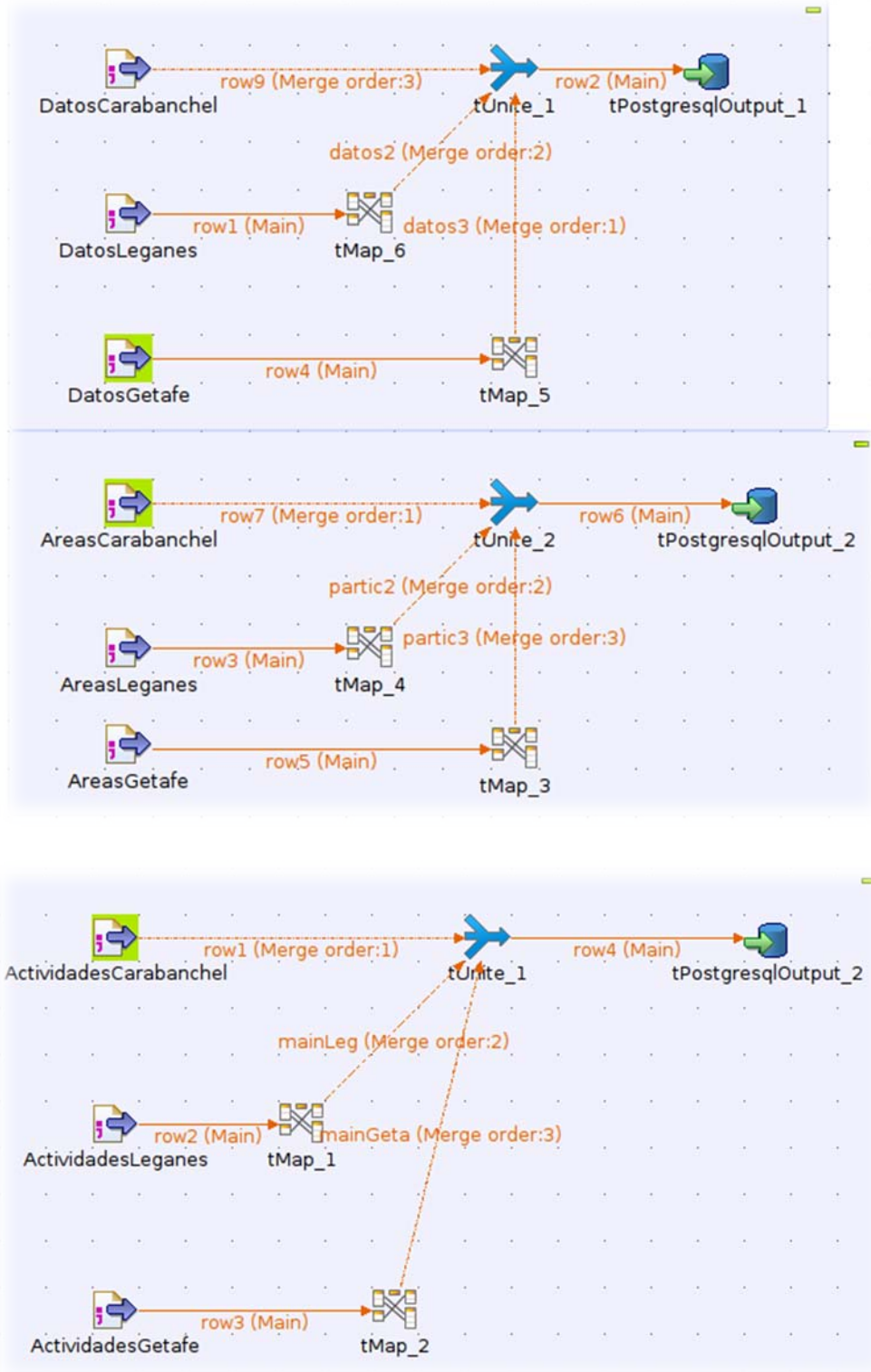
Desde el archivo Excel también:



Limpieza de datos de participantes.

```
Update "Participantes" SET "Nombre"=INITCAP(TRANSLATE(upper(TRIM("Nombre")),
'ÀÁÂÃÄËÊËÎĨÍÎÖÖÓÔÚÜÙ#Ç', 'AAAAEEEEIIIOOOUUUUNC'));Update "Participantes" SET
"PrimerApellido"=INITCAP(TRANSLATE(upper(TRIM("PrimerApellido")), 'ÀÁÂÃÄËÊËÎĨÍÎÖÖÓÔÚÜÙ#Ç',
'AAAAEEEEIIIOOOUUUUNC'));
Update "Participantes" SET
"SegundoApellido"=INITCAP(TRANSLATE(upper(TRIM("SegundoApellido")),
'ÀÁÂÃÄËÊËÎĨÍÎÖÖÓÔÚÜÙ#Ç', 'AAAAEEEEIIIOOOUUUUNC'));
Update "Participantes" SET
"PadreMadreTutor"=INITCAP(TRANSLATE(upper(TRIM("PadreMadreTutor")),
'ÀÁÂÃÄËÊËÎĨÍÎÖÖÓÔÚÜÙ#Ç', 'AAAAEEEEIIIOOOUUUUNC'));
Update "Participantes" SET "Localidad"=INITCAP(TRANSLATE(upper(TRIM("Localidad")),
'ÀÁÂÃÄËÊËÎĨÍÎÖÖÓÔÚÜÙ#Ç', 'AAAAEEEEIIIOOOUUUUNC'));
Update "Participantes" SET "Domicilio"=INITCAP(TRANSLATE(upper(TRIM("Domicilio")),
'ÀÁÂÃÄËÊËÎĨÍÎÖÖÓÔÚÜÙ#Ç', 'AAAAEEEEIIIOOOUUUUNC'));
UPDATE "Participantes" SET "DNI"=TRANSLATE("DNI",'.', '');
UPDATE "Participantes" SET "DNI"='' WHERE "DNI"='00000000T';
UPDATE "Participantes" SET "Telefono"=TRANSLATE("Telefono",'_', '');
UPDATE "Participantes" SET "SegundoTelefono"=TRANSLATE("SegundoTelefono",'_', '');
```

Migras inscripciones y actividades y áreas



Eliminación de Participantes que no han participado en los últimos 4 años:

```
DELETE FROM "Participantes" WHERE
"IdParticipante" in
(SELECT distinct "IdParticipante" FROM "Inscripcion" WHERE "FechaAlta" < '2005-01-01')
and "IdParticipante" not in
(SELECT distinct "IdParticipante" FROM "Inscripcion" WHERE "FechaAlta" >= '2005-01-01')
```

(2391 Registros afectados)

Igualmente se elimina las inscripciones anteriores a esa fecha.

```
DELETE FROM "Inscripcion" WHERE
"FechaAlta" < '2005-01-01'
```

Y las inscripciones de actividades y actividades previas a esa fecha:

```
DELETE from "Inscripcion" WHERE
"IdActividad" in
(Select "IdActividad" from "Actividades"
where "FechaRealizacion" < '2005-01-01')
```

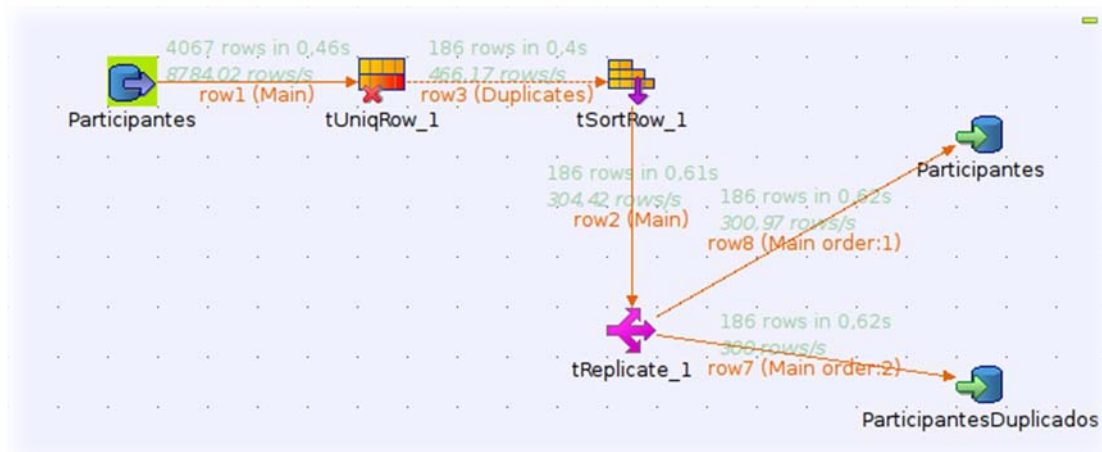
Las actividades:

```
DELETE from "Actividades"
where "FechaRealizacion" < '2005-01-01'
```

Las áreas sin actividades

```
DELETE from "Areas" WHERE
"IdArea" not in
(Select "IdArea" from "Actividades")
```

Se identifican los registros duplicados:



Tras importar todos los datos, se sustituye el id del antiguo participante:

```
UPDATE "Inscripcion" SET "IdParticipante"=orig.newwid FROM (SELECT oldid, newwid
FROM "Duplicados") orig
WHERE "IdParticipante"=orig.oldid
```

Debido a un error en una de las bases de datos, existen registros duplicados dentro de claves primarias, por eso hay que eliminar la restricción de clave única, insertar los cambios y posteriormente eliminar los datos duplicados.

Se limpian a mano las cadenas que son demasiado largas en destino:
length("Email") length("Domicilio")

Paso en el servidor:

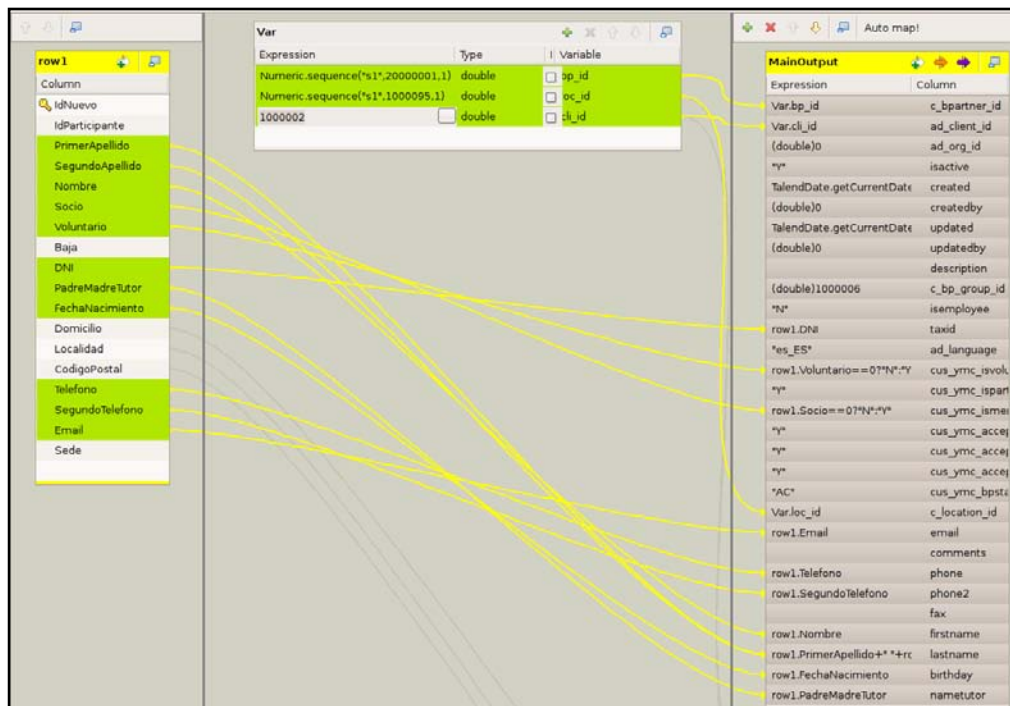
Para migrar los datos de los usuarios hay que importar dos tablas, c_location y c_bpartner (esta última a través de su vista cus_ymc_bpartner_v)

Para mantener el id de secuencia de la aplicación es necesario conocer el valor siguiente de secuencia de la tabla cus_ymc_bpartner_v y el de c_location, por medio de las siguientes sentencias:

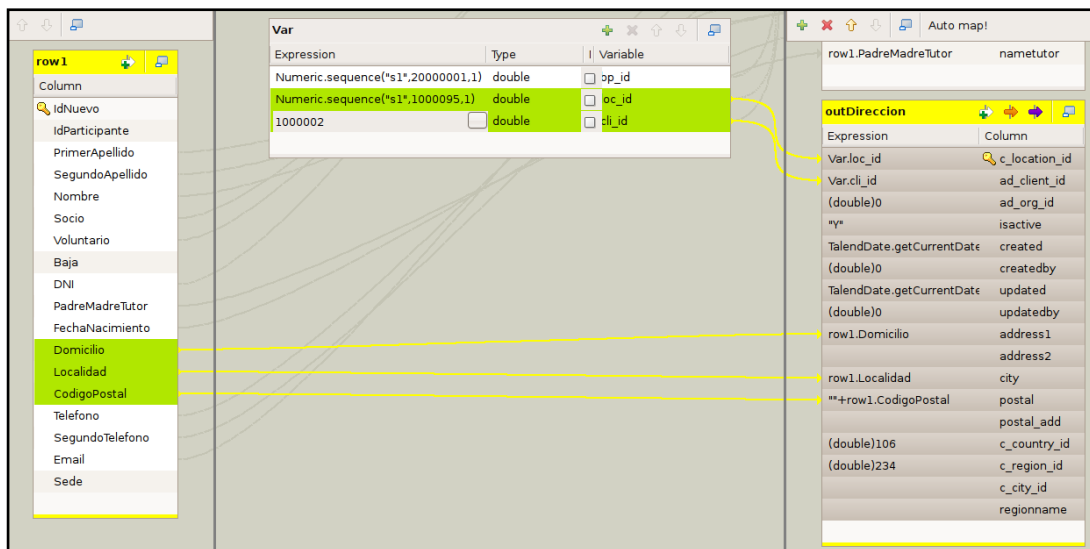
```
SELECT currentnext, name FROM ad_sequence where
name='C_Location' or name='cus_ymc_bpartner_v'

20000001; "cus_ymc_bpartner_v"
10000095; "C_Location"
```

Así se establecen los valores de las variables temporales bp_id y loc_id (Que se muestran en la columna intermedia).



Este es el mapeo que se hace a la tabla c_location.

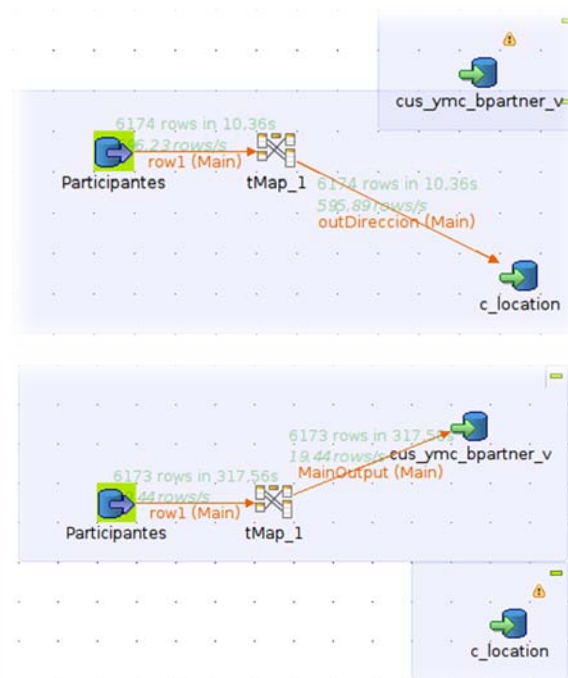


Var

Expression	Type	Variable
Numeric.sequence(*s1*,20000001,1)	double	bp_id
Numeric.sequence(*s1*,1000095,1)	double	pc_id
1000002	double	cli_id

Auto map!

Expression	Column
row1.PadreMadreTutor	nametutor
Var.loc_id	c_location_id
Var.cli_id	ad_client_id
(double)0	ad_org_id
"y"	isactive
TalendDate.getCurrentDate	created
(double)0	createdby
TalendDate.getCurrentDate	updated
(double)0	updatedby
row1.Domicilio	address1
row1.Localidad	address2
row1.Localidad	city
""+row1.CodigoPostal	postal
(double)106	postal_add
(double)106	c_country_id
(double)234	c_region_id
	c_city_id
	regionname



Se rectifica el modificador de la vista `cus_ymc_bpartner_v` para la insercción:

```
DROP RULE cus_ymc_bpartner_v_ins ON cus_ymc_bpartner_v;

CREATE OR REPLACE RULE cus_ymc_bpartner_v_ins AS
  ON INSERT TO cus_ymc_bpartner_v DO INSTEAD ( INSERT INTO c_bpartner (c_bpartner_id,
ad_client_id, ad_org_id, isactive, created, createdby, updated, updatedby, value, name,
description, c_bp_group_id, isemployee, ad_language, taxid, cus_ymc_isvolunteer,
cus_ymc_isparticipant, picture_url, cus_ymc_ismember, cus_ymc_acceptsms, cus_ymc_acceptmail,
cus_ymc_acceptemail, cus_ymc_bpstatus)
  VALUES (new.c_bpartner_id, new.ad_client_id, new.ad_org_id, new.isactive, new.created,
new.createdby, new.updated, new.updatedby, new.c_bpartner_id, (((new.firstname::text || '
::text) || new.lastname::text)), new.description, new.c_bp_group_id, new.isemployee,
new.ad_language, new.taxid, new.cus_ymc_isvolunteer, new.cus_ymc_isparticipant,
new.picture_url, new.cus_ymc_ismember, new.cus_ymc_acceptsms, new.cus_ymc_acceptmail,
new.cus_ymc_acceptemail, new.cus_ymc_bpstatus);
  INSERT INTO c_bpartner_location (c_bpartner_location_id, ad_client_id, ad_org_id, isactive,
created, createdby, updated, updatedby, c_bpartner_id, c_location_id, name)
  VALUES (ad_sequence_next('C_BPartner_Location'::character varying, new.ad_client_id),
new.ad_client_id, new.ad_org_id, new.isactive, new.created, new.createdby, new.updated,
new.updatedby, new.c_bpartner_id, new.c_location_id, substring((( SELECT (('::text ||
COALESCE(c_location.city::text, '::text)) || ', '::text) ||
COALESCE(c_location.address1::text, '::text) AS name
FROM c_location
WHERE c_location.c_location_id = new.c_location_id))) from 1 for 60 ));
  INSERT INTO ad_user (ad_user_id, ad_client_id, ad_org_id, isactive, created, createdby,
updated, updatedby, name, email, c_bpartner_id, c_bpartner_location_id, c_greeting_id,
title, comments, phone, phone2, fax, birthday, firstname, lastname, username, nametutor)
  VALUES (ad_sequence_next('AD_User'::character varying, new.ad_client_id),
new.ad_client_id, new.ad_org_id, new.isactive, new.created, new.createdby, new.updated,
new.updatedby, (((new.firstname::text || ' '::text) || new.lastname::text)), new.email,
new.c_bpartner_id, (( SELECT ad_sequence.currentnext - ad_sequence.incrementno
FROM ad_sequence
WHERE ad_sequence.name::text = 'C_BPartner_Location'::text)), new.c_greeting_id,
new.title, new.comments, new.phone, new.phone2, new.fax, new.birthday, new.firstname,
new.lastname, NULL::character varying, new.nametutor);
);
```

Este problema desembocaría en un fallo.

Establecer los nuevos identificadores en la tabla `ad_sequence`

```
select max(c_bpartner_id) from c_bpartner --> 20003880 + 1 = 20003881

UPDATE ad_sequence SET currentnext=20003881 where
name='cus_ymc_bpartner_v'

select max(c_location_id) from c_location --> 1003957 + 1 = 1003958

UPDATE ad_sequence SET currentnext=1003958 where
name='C_Location'
```

Cargar los datos de actividades:

Se hace la asignación de Áreas de forma manual,

Lo que antes eran Áreas ahora serán Proyectos.

Lo que antes era Actividades ahora serán Programas.

La relación Áreas - Actividades - Participantes existe que se traducirá a Proyecto - Programa – InscripciónBPParticipantes.

Normalización de Áreas y Actividades:

```
update "Areas" set "Area" = initcap("Area")
update "Actividades" set "Actividad" = initcap("Actividad")
```

Se amplía la información de Áreas, añadiendo la fecha de comienzo a partir de la primera actividad inscrita.

```
ALTER TABLE "Areas" ADD COLUMN "FechaRealizacion" date;
ALTER TABLE "Areas" ALTER COLUMN "FechaRealizacion" SET STORAGE PLAIN;

Update "Areas" SET "FechaRealizacion"=b.fecha from
(Select area."IdArea" as idarea, area."Area", min(acti."FechaRealizacion")
as fecha from "Areas" area
join "Actividades" acti on area."IdArea"=acti."IdArea"
group by area."IdArea", area."Area") b
where "IdArea"=b.idarea
```

Se revisan a mano las áreas y se asigna el id del Área de la aplicación en la que se anidará. Siendo:

Área	Id
Infancia	1000002
Juventud	1000003
Adultos	1000004
Varios	1000005

```
ALTER TABLE "Areas" ADD COLUMN "AreaAplicacion" int;
```

Unificar Áreas comunes en diferentes sedes.

Reemplazar	Por
30013	60
30011, 20034	61
30003	20007
30002	20016
20032	18
20033	31

```
update "Actividades" set "IdArea"=60 where "IdArea"=30013
```

Eliminación manual de algunas Áreas

Eliminación de Actividades si Área asociada.

```
DELETE from "Actividades" where "IdArea" not in  
(Select "IdArea" from "Areas")
```

Eliminación de las Inscripciones sin Actividades asociadas.

```
delete from "Inscripcion" where "IdActividad" not in  
(Select "IdActividad" from "Actividades")
```

Eliminar Actividades que no han sido inscritas nunca

```
Delete from "Actividades" where "IdActividad" not in  
(Select "IdActividad" from "Inscripcion")
```

Eliminación de las Áreas huérfanas

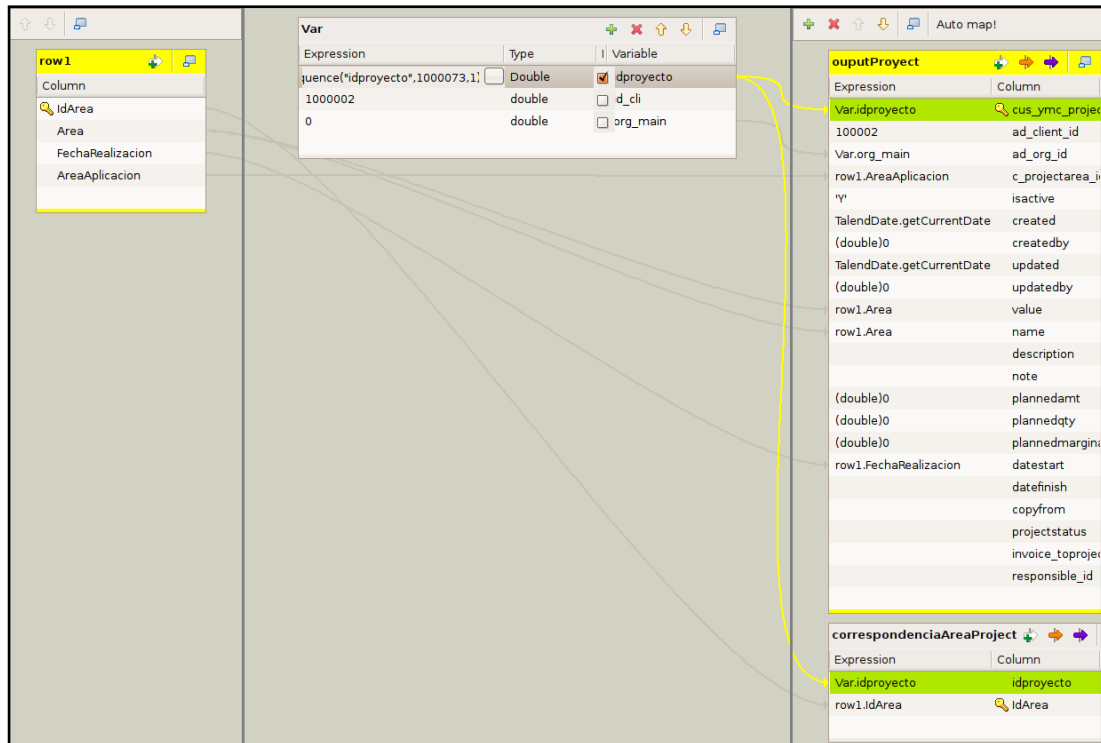
```
DELETE from "Areas" where "IdArea" not in  
(Select "IdArea" from "Actividades")
```

Migración al servidor:

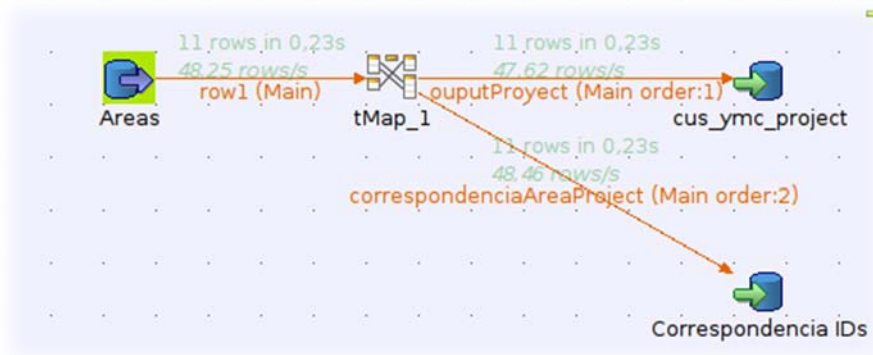
Se crea un m apping entre la tabla Áreas y cus_ymc_project usando el valor actual del id del proyecto:

```
SELECT currentnext, name FROM ad_sequence where  
name='CUS_YMC_PROJECT' or name='CUS_YMC_PROGRAM'  
or name='CUS_YMC_PARTICIPATE_BP_PROG'
```

ID	Nombre
1000003	CUS_YMC_PROGRAM
1000006	CUS_YMC_PROJECT
1000073	CUS_YMC_PARTICIPATE_BP_PROG



Añadimos además una tabla con la correspondencia entre el id asignado en la aplicación y el que tenía en la aplicación original.

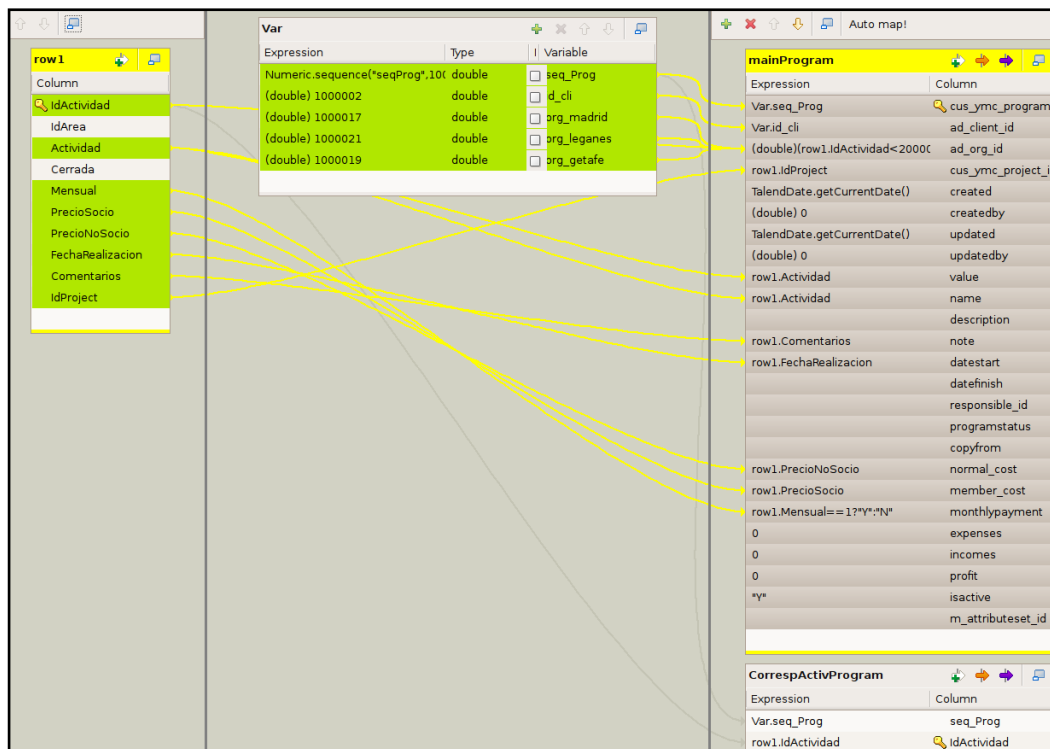


Usar tabla de correspondencia para cambiar id's de áreas de actividades.

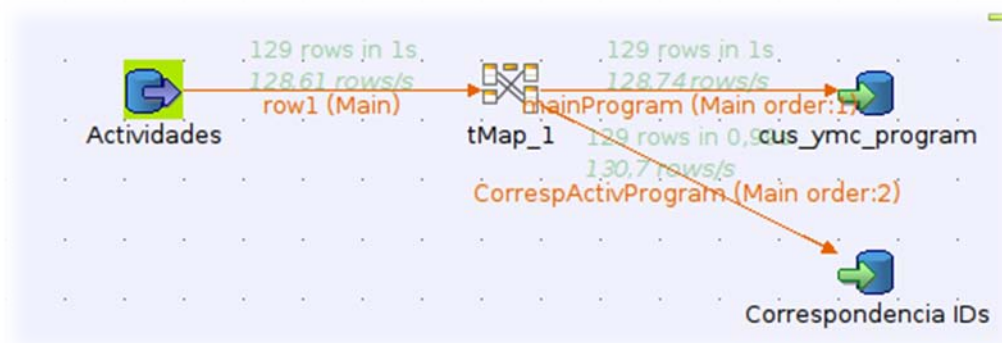
```
ALTER TABLE "Actividades" ADD COLUMN "IdProject" integer;

update "Actividades" set "IdProject"=b."idproject" from
(select "IdArea" as area, "null" as "idproject"
 from "CorrespondenciaAreasProject") b
where "IdArea"=b.area
```

Se crea la correspondencia entre la tabla origen Actividades y la destino cus_ymc_program



Manteniendo la relación del origen de actividad y agregando una tabla de salida con la relación entre el id asignado y el id que tenía cada actividad, para la posterior corrección en la tabla de inscripciones.



Modificamos la tabla Inscripción para añadir los campos de correspondencia:

```
ALTER TABLE "Inscripcion" ADD COLUMN "IdProgram" integer;
ALTER TABLE "Inscripcion" ADD COLUMN "IdBPartner" integer;
ALTER TABLE "Inscripcion" ADD COLUMN "Coste" float;
```

Actualización del id del participante:

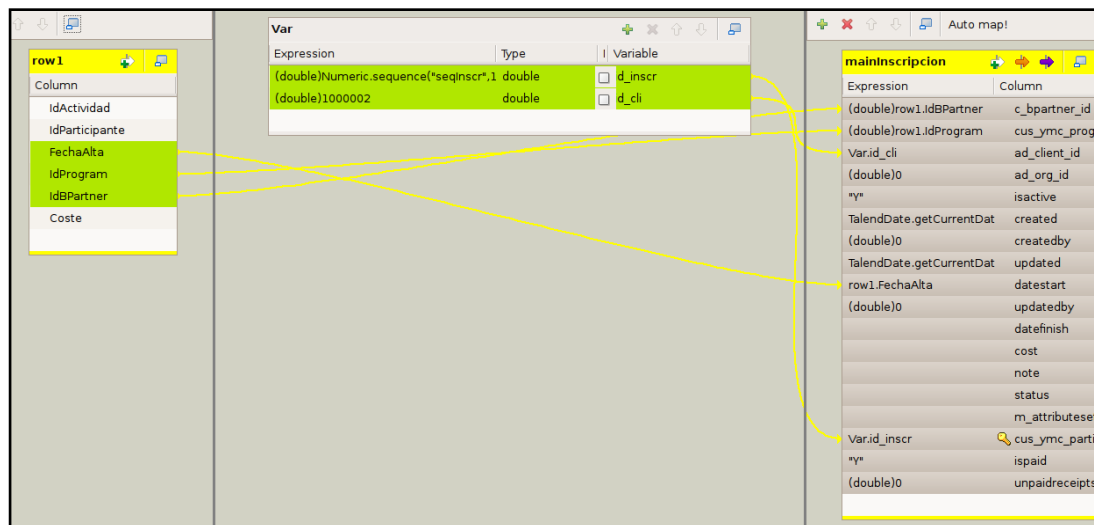
```
update "Inscripcion" set "IdBPartner"=b."idbp" from
(Select "IdParticipante" as participante, "null" as "idbp"
from "Correspondencia") b
```

```
where "IdParticipante"=b.participante
```

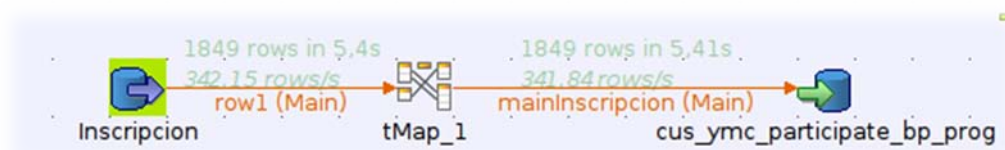
Actualización del id de la actividad:

```
Update "Inscripcion" set "IdProgram"=b."idprog" from
(Select "IdActividad" as actividad, "null" as "idprog"
 from "CorrespondenciaActiviProgram") b
where "IdActividad"=b.actividad
```

Se crea la correspondencia entre la tabla origen inscripción y la tabla destino
cus_ymc_participate_bp_prog



Y se ejecuta:



A continuación se actualizan los identificadores de cada tipo de registro:

```
select max(cus_ymc_participate_bp_prog_id) from cus_ymc_participate_bp_prog;

update ad_sequence set currentnext=1001922
where name='CUS_YMC_PARTICIPATE_BP_PROG';

select max(cus_ymc_program_id) from cus_ymc_program;

update ad_sequence set currentnext=1000132
where name='CUS_YMC_PROGRAM';

select max(cus_ymc_project_id) from cus_ymc_project;

update ad_sequence set currentnext=1000084
where name='CUS_YMC_PROJECT';
```

Se indica el precio a pagar en cada inscripción:

```
update cus_ymc_participate_bp_prog
set "cost"=b.normal_cost from
(select cus_ymc_program_id as progid, normal_cost from cus_ymc_program
where ad_client_id=1000002) b
where cus_ymc_program_id=b.progid
```

Se pone fecha límite a los programas con el siguiente criterio:

Actividades anteriores a 2008 fecha de finalización es el mismo año a 31 de diciembre:

```
UPDATE cus_ymc_program set datefinish='2005-12-31'
where ad_client_id=1000002
and datestart > '2005-01-01' and datestart < '2006-01-01';

UPDATE cus_ymc_program set datefinish='2006-12-31'
where ad_client_id=1000002
and datestart > '2006-01-01' and datestart < '2007-01-01';

UPDATE cus_ymc_program set datefinish='2007-12-31'
where ad_client_id=1000002
and datestart >= '2007-01-01' and datestart < '2008-01-01';
```

Toda actividad de 2008 anterior a Junio termina en Junio.

```
UPDATE cus_ymc_program set datefinish='2008-05-31'
where ad_client_id=1000002
and datestart >= '2008-01-01' and datestart < '2008-06-01'
```

Toda actividad del verano de 2008 termina en septiembre.

```
UPDATE cus_ymc_program set datefinish='2008-09-10'
where ad_client_id=1000002
and datestart >= '2008-06-01' and datestart < '2008-09-10'
```

Toda actividad de 2008/2009 queda abierta (definir fechas de forma manual).

Poner fecha de fin a todas las inscripciones en programas ya finalizados:

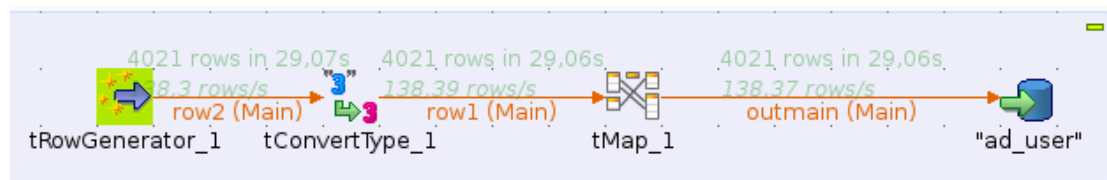
```
Update cus_ymc_participate_bp_prog
set datefinish=b.dfin from
(select cus_ymc_program_id as id, datefinish as dfin from cus_ymc_program
where datefinish<='10-09-2008') b
where cus_ymc_program_id=b.id
```

2 Migración al sistema demostrativo

Como parte del tratamiento de datos del proyecto se encuentra también el proceso de carga de datos del sistema demostrativo que se entrega junto a este proyecto.

Se pretendía contar con un sistema lo más fiel posible para demostrar todas las características del sistema con datos casi reales. Por ello la carga inicial del sistema parte de los datos reales del sistema productivo.

Para garantizar la confidencialidad de los datos personales de los participantes y trabajadores se somete el conjunto de datos a una serie de transformaciones.



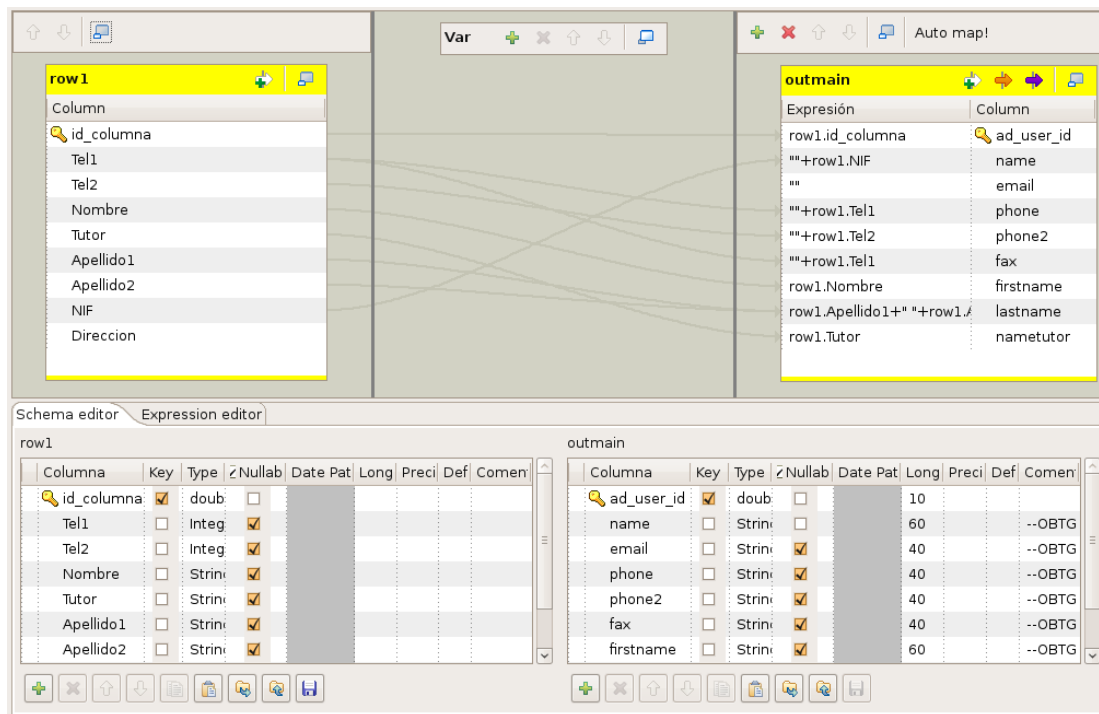
Captura 52: Reemplazo de datos personales

De nuevo en esta fase se utiliza la herramienta Talend Studio, ya que nos proporciona otra herramienta de utilidad en este proceso que es la generación de datos específicos de forma aleatoria. Con esta utilidad se generan datos que sustituirán nombre, apellidos, teléfonos, direcciones y números identificativos de todos los registros personales.

Schema							Functions	
Columna	Key	Type	Nullable	Longitud	Precision	Default	Functions	Parámetros
phone	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>				random	min value=>5000
phone2	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>				random	min value=>5000
Nombre	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>				getFirstName	
Tutor	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>				getFirstName	
Apellidos	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>				getLastName	
NIF	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>				random	min value=>1000
Direccion	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>				getUsStreet	

Captura 53: Generación de datos ficticios

El resto de tablas que muestran datos personales, se sustituyen de igual modo, con Talend Studio o a partir de los datos reemplazados en esta tabla, con ayuda de sentencias SQL.



Captura 54: Mapeo de datos ficticios









Además se eliminan todas las contraseñas y usuarios existentes en el sistema original (incluyendo el del servicio de envío de SMS), y se crea un usuario con privilegios y varios roles que es el usuario 'demo'.

Anexo III - Contenido del DVD

1 Contenido del DVD

En la Tabla 7 se muestra el contenido del DVD incluido en este proyecto.

Tabla 7: Contenido del DVD

Archivo	Descripción
 Léame.txt Documento de texto 1 KB	Archivo con descripción del contenido del DVD
Documentación Proyecto	
 ProyectoYMCAOpenbravo.docx Documento de Microsoft Office...	Memoria del proyecto en su formato original.
 ProyectoYMCAOpenbravo.pdf Adobe Acrobat Document	Memoria del proyecto en formato PDF.
Curso Online	
 CursoYobScorm.zip 36.256 KB	Archivo SCORM del curso de la plataforma YMCA Openbravo, listo para ser desplegado en un gestor de aprendizaje.
 CursoYob	Carpeta con curso online disponible para ser visualizado desde el navegador sin gestor de aprendizaje.
Máquina Virtual	
 YMCA OpenbravoERP-2.40-x86.zip 439.413 KB	Archivo comprimido con la imagen de la máquina virtual demostrativa del sistema.
 VMware-player-2.5.2-156735.... VMware installation launcher VMware, Inc.	Este programa permite ejecutar máquinas virtuales construidas con VMware Workstation, VMware distribuye esta aplicación de forma gratuita y puede ser descargada desde: http://www.vmware.com/download/player/download.html
Código Fuente	
 Fuente YMCA Openbravo.zip 91.406 KB	Código fuente del proyecto, desde este archivo es posible desplegar el sistema utilizando los scripts de ant.

2 Ejecución del entorno demostrativo

En esta sección se describe cómo ejecutar el entorno demostrativo disponible en el DVD.

El sistema desarrollado durante este proyecto, no necesita de un entorno preconfigurado, con una base de datos, un servidor de aplicaciones y un servidor web, además de versiones específicas de algunas librerías. Para facilitar las pruebas de la aplicación se ha creado una máquina virtual, con todo el sistema desplegado. El programa capaz de ejecutar esta máquina virtual es VMWare Player que se incluye con el DVD.

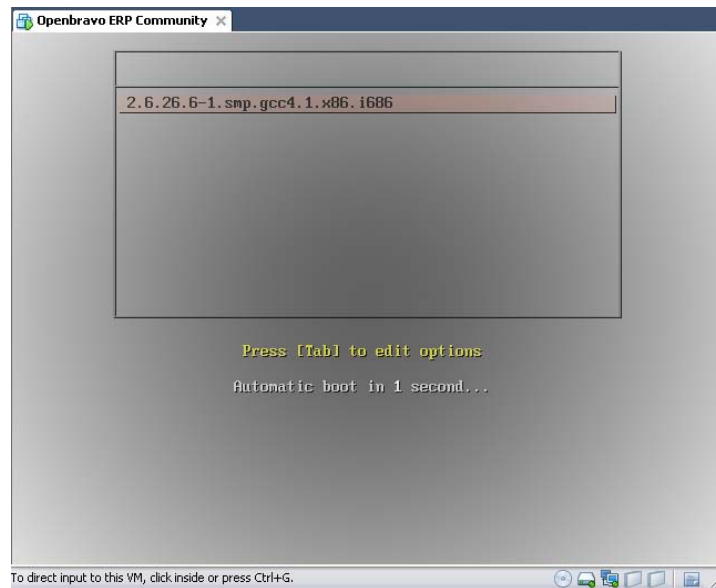
El primer paso será instalar la aplicación VMWare Player, si no se dispone ya de ella, o de su equivalente VMWare Workstation.

A continuación es necesario descomprimir el archivo `YMCA_OpenbravoERP-2.40-x86.zip` en el disco duro o en una unidad extraíble, que tenga una capacidad de al menos 1,7 Gb, que es lo que ocupa la imagen descomprimida y los archivos que genera durante su arranque.



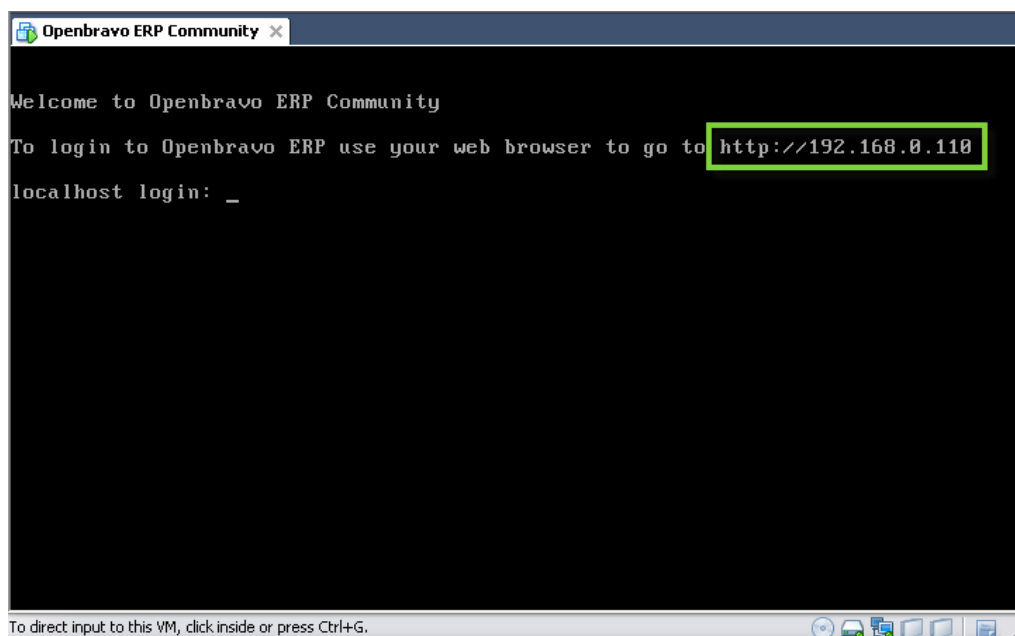
Captura 55: Ficheros de la máquina virtual

Como resultado tendremos 2 archivos, uno que contiene la máquina virtual, y otro que define las características de la máquina virtual, arrancaremos el sistema haciendo doble click sobre la segunda (`OpenbravoERP-2.40-x86.vmx`).



Captura 56: Arranque de máquina virtual

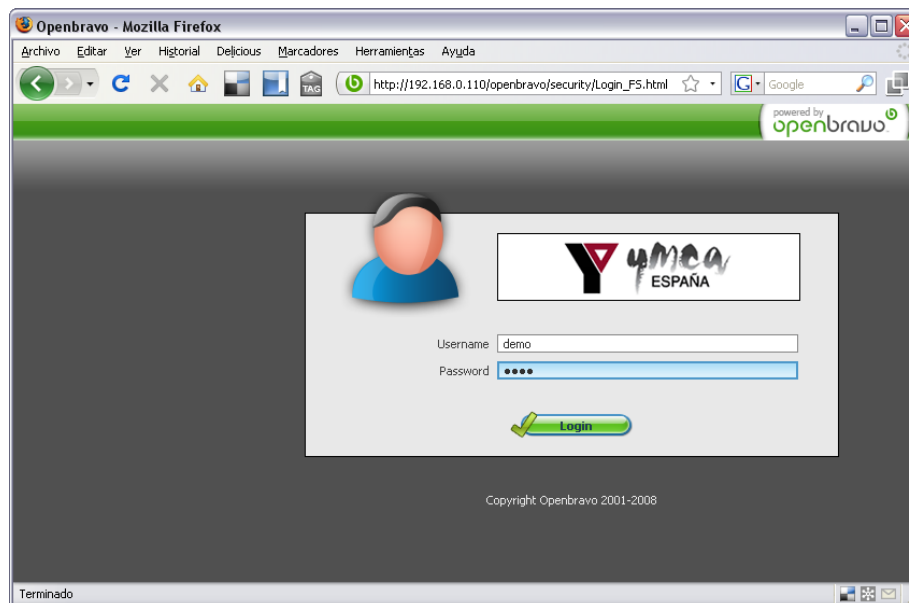
Una vez arrancado el sistema se mostrará la pantalla mostrada en la Captura 57, que además de mostrar el prompt para hacer login, nos muestra la configuración actual que nos permitiría arrancar la aplicación desde nuestro navegador.



Captura 57: Ventana de Login de la máquina virtual

Los datos para acceder al sistema, que nos permitirá apagar la máquina (comando `halt`) entre otras cosas, son los siguientes:

Usuario:	root
Password:	openbravo

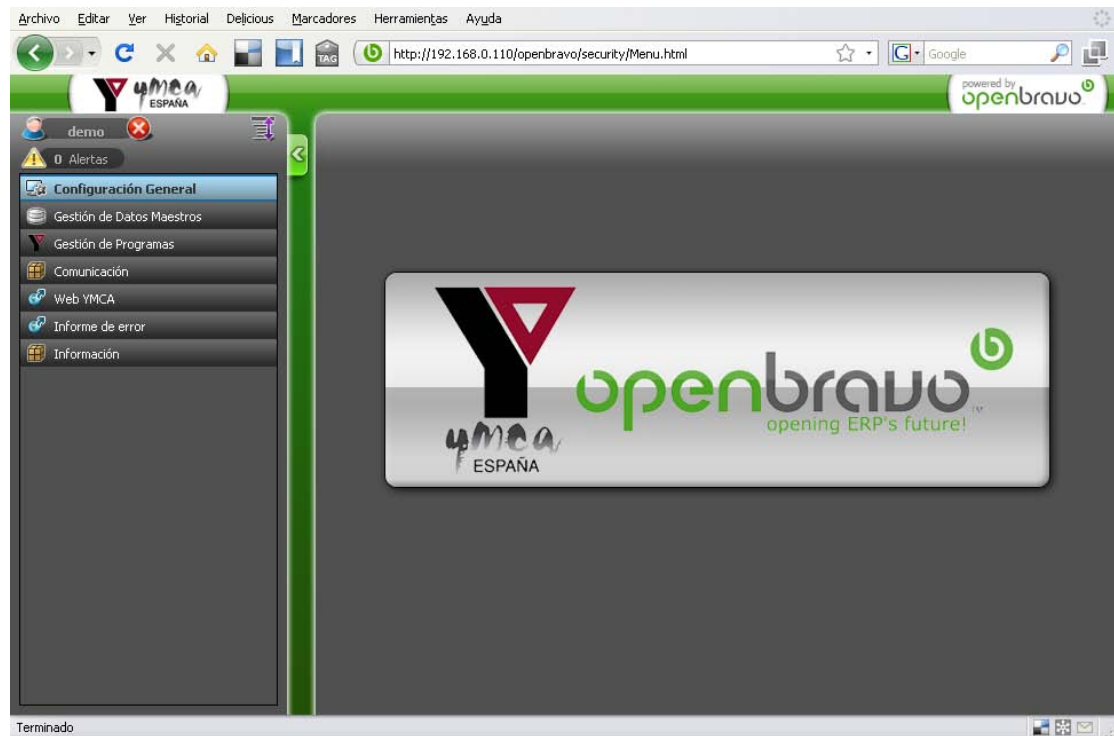


Captura 58: Acceso a la aplicación

El usuario de pruebas creado tiene las siguientes credenciales.

Usuario:	demo
Password:	demo

La información de los participantes contenida en la aplicación de pruebas es ficticia (tratamiento de datos descrito en la sección 2 del Anexo II) para garantizar la seguridad de los datos de la entidad propietaria.



Captura 59: Login en la aplicación con el usuario demo

Al tratarse de un sistema ficticio, es posible hacer tantas pruebas como se quiera, y bastaría volver a descomprimir el archivo de la imagen virtual para recuperar el estado original del sistema.

3 Curso online

El curso online ha sido diseñado para ser desplegado en la plataforma de aprendizaje de la organización.


Además se ha generado una carpeta que contiene todo el curso listo para ser visualizado desde el navegador sin la ayuda de un gestor de aprendizaje. Para visualizar el curso hay que acceder a la carpeta "Curso Online/CursoYob" del DVD, localizar el archivo index.html y abrirlo desde el navegador web.

Curso YMCA Openbravo

[Inicio](#)
Organización del Curso
[¿Por qué YMCA Openbravo?](#)
[Introducción a YMCA Openbravo](#)
[Procesos habituales](#)
[Herramientas Administrativas](#)
[Reporte de Errores](#)
[Ampliando el conocimiento](#)

Organización del Curso

El presente curso ha sido creado para proporcionar un primer contacto de los técnicos de YMCA a la futura aplicación de gestión de la entidad YMCA Openbravo.



Objetivos

El curso ha sido concebido para alcanzar los siguientes objetivos:

- Conocer los elementos principales de la herramienta YMCA Openbravo para ser capaces de usarla con soltura.
- Ser capaz de realizar las operaciones habituales en el día a día de la asociación.
- Conocer el procedimiento que se ha establecido para reportar errores y solicitar nuevas funcionalidades.
- Conocer y utilizar los medios se han establecido para ampliar el conocimiento de la aplicación.

Como parte del curso se realizará una serie de ejercicios que tienen como objetivo el asentar las descripciones realizadas de los procesos habituales.

Al comienzo de cada ejercicio se proporciona un objetivo y una descripción general de la tarea. Se recomienda realizar el ejercicio sin seguir al pie de la letra las descripciones, usando estas sólo como método orientativo si la intuición no nos orienta adecuadamente.

Ilustración 46: Curso Online sin entorno de aprendizaje

Visualizar este curso es una buena forma de conocer el sistema y verlo funcionando (a través de los videos), sin necesidad de ejecutarlo desde la máquina virtual.